CCM Processor Users' Guide
(UNICOS Version)

LAWRENCE E. BUJA
PREFACE

This version of the CCM Processor Users' Guide (UNICOS Version) supercedes all previous versions (NCAR/TN-290+IA, NCAR/TN-335+IA), updates and drafts of this document.

I would like to thank all of the CCM users and programmers who have contributed to the development of the CCM Processor over the years since its inception. Rick Wolski was the primary developer of both the Processor code and the extensive documentation which accompanies it. Tom Mayer is the author of all of the spectral processing code. He was also the primary implementer of the options for isentropic vertical interpolation and the output of LSD Save Tapes.

Lawrence Buja
November 1993
### TABLE OF CONTENTS

**Part I. Description of General Processor Capabilities** ................ I-1

1. **DESC** - Brief description of the Processor .................. I-1
2. **INTR** - Introduction to the documentation .................. I-2
   2.1. **DEFS** - Definitions of Processor Terms and Acronyms .... I-5

3. **USPR** - Using the Processor .................................. I-9
   3.1. **CJDK** - The Cray job deck ................................ I-9
   3.1.1. **JOBS** - Jobstepping .................................. I-13
   3.1.2. **SAMP** - Accessing the sample Processor job scripts ... I-14
3.2. **NICP** - Naming conventions for ICP keywords ............. I-15
3.3. **SICP** - Syntax of ICP data input records .................. I-15
3.4. **MSIN** - Mass storage interactions .......................... I-16
3.5. **ERRS** - Error handling ..................................... I-20
3.6. **ORDR** - Order of Processor operations ..................... I-22
   3.6.1. **ORNN** - Order for no time averaging, no time filtering I-23
   3.6.2. **ORAN** - Order for time averaging, no time filtering .... I-23
   3.6.3. **ORNF** - Order for no time averaging, time filtering ... I-24
   3.6.4. **ORAF** - Order for time averaging, time filtering ....... I-25
   3.6.5. **ORSP** - Order of spectral operations ................... I-25
   3.6.6. **ORSA** - Order of spatial averaging and related plotting I-26
   3.6.7. **ORPC** - Order of Case C processing ..................... I-26
   3.6.8. **ORST** - Order of Save Tape output ...................... I-26

3.7. **SPCL** - Special topics ...................................... I-27
   3.7.1. **MLIM** - Memory limitations ................................ I-27
   3.7.2. **COST** - Processing costs .................................. I-28
   3.7.3. **CHEM** - Conserving memory ................................ I-30
   3.7.5. **XBTL** - Extra bottom (surface) levels ................... I-32
   3.7.6. **MIXR** - Special contour levels for mixing ratio ....... I-32

4. **DATA** - Specification of data to process ...................... I-33
4.1. **IHST** - Input of History tapes ............................ I-33
   4.1.1. **LINL** - Limiting history tape input levels ............. I-34

4.2. **ENSB** - Ensemble processing .................................. I-34
4.3. **LSDP** - List sorted data (LSD) processing .................. I-35
4.4. **PRNT** - Printout of field values ........................... I-36
4.5. **SAVT** - Save tapes ........................................ I-37
   4.5.1. **STAV** - Time Average Save Tapes ....................... I-38
   4.5.1.1. **ITAV** - Input of Time Average Save Tapes ............ I-38
   4.5.1.2. **OTAV** - Output of Time Average Save Tapes .......... I-39

4.5.2. **STSR** - Time Series Save Tapes .......................... I-39
   4.5.2.1. **ITSR** - Input of Time Series Save Tapes ............. I-40
   4.5.2.2. **OTSR** - Output of Time Series Save Tapes .......... I-40

4.5.3. **SHST** - History Save Tapes ................................ I-41
   4.5.3.1. **OHST** - Output of History Save Tapes .............. I-41
4.5.4. SHSL - Horizontal Slice Save Tapes ......................... I-43
4.5.4.1. OHOR - Output of Horizontal Slice Save Tapes .......... I-43

4.5.5. SFCT - Surface Type Save Tapes .......................... I-45
4.5.5.1. ISFT - Input of Surface Type Save Tapes ................. I-45
4.5.5.2. OSFT - Output of Surface Type Save Tapes .............. I-45

4.5.6. STSP - Time Series Plot Save Tapes ....................... I-45
4.5.6.1. ITSP - Input of Time Series Plot Save Tapes .......... I-46
4.5.6.2. OTSP - Output of Time Series Plot Save Tapes .......... I-46

4.5.7. LSDS - List Sorted Data Save Tapes ....................... I-47
4.5.7.1. OLSD - Output of LSD Save Tapes ...................... I-47

5. DFLD - Derived fields ........................................... I-54
5.1. GDFL - General description of derived fields .............. I-54
5.2. CDFL - Code-defined derived fields ........................ I-57
5.3. UDFL - User-defined derived fields ......................... I-76
5.4. SDFL - Spectral derived fields ............................ I-77
5.4.1. VPDF - Vector-pair derived fields ...................... I-77
5.4.2. GRDF - Gradient derived fields ........................ I-78

6. VERT - Vertical interpolation .................................. I-78
6.1. VPRS - Vertical interpolation to pressure surfaces .... I-78
6.2. VPOT - Vertical interpolation to potential temperature surfaces I-80

7. SPEC - Spectral processing .................................... I-81
7.1. SINT - Spectral interpolation ............................... I-82
7.2. SBND - Spectral bandpass filtering ......................... I-83

8. TAVG - Time average statistics ................................ I-84
9. COMP - Case comparison ........................................ I-85
10. CMRG - Case merging ........................................... I-87
11. MASK - Surface type masking .................................. I-87
12. TFIL - Time filtering ........................................ I-88
14. ZEST - Zonal eddy statistics ................................ I-89
14.1. VERA - Vertical averaging .................................. I-90
14.2. ZONA - Zonal averaging .................................... I-91
14.3. MERA - Meridional averaging ............................... I-92
14.4. HORA - Horizontal area averaging with surface type masking I-92

15. PLOT - Production of graphics ................................ I-94
15.1. PDIS - Disposition of plot file ............................ I-95
15.2. PHOR - Plotting horizontal projections ................. I-97
15.2.1. PHRE - Plotting rectangular horizontal projections .... I-97
15.2.1.1. LAHP - Limited area horizontal projections .......... I-98
15.2.2. PHPO - Plotting polar horizontal projections .......... I-99

15.3. PMEX - Plotting meridional cross sections .............. I-100
15.4. PLAX - Plotting latitudinal cross sections ............. I-101
15.5. PTIM - Plotting time series ............................... I-102
November 1993

15.5.1. PTFL - Plotting time vs. field value ........................................... I-103
15.5.2. PTZL - Plotting time vs. zonal average ....................................... I-103
15.5.3. PTAL - Plotting time vs. masked area average ................................ I-104
15.5.4. PTZC - Plotting zonal avg. contours (time vs. latitude) ................ I-105
15.5.5. PTMC - Plotting meridional avg. contours (long. vs. time) ............ I-105

15.6. SPGR - Spectral graphics ........................................... I-106
15.7. PVAL - Plotting printed values ............................................... I-106
15.8. PLIN - Plotting line graphs ............................................. I-107
15.9. PCON - Plotting contours ................................................ I-107
15.10. PVEC - Plotting vectors ................................................ I-108
15.11. PCLR - Plotting in color ............................................... I-109

Part II. Description of Specific Input Control Parameters (ICPs) ........ II-1

Appendix A - Sample plots .......................................................... A-1

1. Rectangular Horizontal Projection Contours .................................. A-1
2. Surface Type As A Printed Value Section .................................... A-2
3. Polar Projection Contours ................................................ A-3
4. Polar Projection Vectors .................................................. A-4
5. Rectangular Horizontal Projection Vectors .................................. A-5
6. Meridional Cross-Section Contours .......................................... A-6
7. Zonal Average Line Plot ................................................... A-7
8. Latitudinal Cross-Section Contours .......................................... A-8
9. Spectral Plot ................................................................. A-9
10. Point Value Time Series Line Plot .......................................... A-10
11. Zonal Average Time Series Line Plot ...................................... A-11
12. Multiple Area Average Time Series Line Plot ............................ A-12
13. Hovmoller Meridional Average Time Series Contours ..................... A-14
14. Zonal Average Time Series Contours ...................................... A-14

Appendix B - Examples of Some Complex Processing Techniques ........ B-1

1. Creating CCM Initial Data Tapes ........................................ B-1
2. Unblocking by Vertical Extrapolation ..................................... B-3
3. Vertical Integration with CCM1 data ..................................... B-5
4. Vertical Finite Differences ............................................... B-7
5. Time and Space Correlation Statistics ................................... B-8
6. Significance Statistics .................................................. B-13
7. Event Statistics .......................................................... B-14
8. Complex Time Average Statistics ........................................ B-16
9. Meridional Streamfunction ............................................... B-17
10. Isentropic Potential Vorticity ......................................... B-19
11. FORTRAN Code for Reading CCM2 History Tapes ...................... B-21
Part I. Descriptions of General Processor Capabilities

1.1 - Brief description of the Processor

The NCAR CCM Processor is the primary tool for the analysis of data output by the NCAR Community Climate Model (CCM). Its main function is to perform the processing and data reduction normally associated with Model climate runs. It is also useful for the analysis of forecast and observational data that are in history tape format. Like the CCM, the Processor is a community facility maintained by the CCM Core Group.

The Processor is designed to be run in batch mode on the Cray computers at NCAR. Model data is initially input in the form of CCM history tapes, and output may be in the form of graphics, printed values, and binary data written to a number of different types of save files (some of which may also be used as subsequent input). A wide range of data manipulation and statistical computation options are available. Processor instructions are given in the form of Input Control Parameters (ICPs). ICP syntax is described in section 1.3.3.

Each major processing capability is isolated in a group of closely related subprograms referred to as a Module. A Module typically accepts a "unit" of data as input, operates on the data, and produces an output "unit". These data units exist independently of any Module; they are referenced through a simple database manager designed specifically for the Processor. 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, i.e., the output of one Module becomes the input of the next. Other Modules, such as those which do spectral operations, may branch off and parallel the main data stream. Which Modules are executed is determined by the values assigned to input parameters with simple statements contained in a file within the run deck. Similar input parameters control options within the various Modules. Virtually all input parameters default to a predetermined setting if they are not specified by the user, and all major processing operations except for the reading of input data are skipped by default. Although the order in which the Modules are executed is fixed within the program, it is possible to effectively reorder operations by chaining together several executions of the Processor so that the output of one execution becomes the input for the next (see section 3.1.1. JOBS - Jobstepping).

All data manipulations are done in main memory, resulting in a minimal amount of I/O and very high computational efficiency. Memory is managed dynamically (i.e., the workspace used by one Module is re-used by the next in the series), so a very large number of Modules can be strung together in the same run. Dynamic memory management also allows the Model configuration parameters (e.g., space and time resolution) to be handled automatically without wasting memory.

One of the most powerful capabilities of the Processor is its ability to
compute new diagnostic quantities based on input or previously computed quantities. These "derived fields" may be defined by the user with arithmetic expressions assigned to an input parameter, so code modifications are usually not required.

A standard run deck is available for executing the Processor on the Crays at NCAR (see section I.3.1.2, SAMP). Since all input and output datasets are handled by the Processor code during execution, this deck requires virtually no modification of the Cray UNICOS commands. There is also a provision for multiple executions of the code within a single Cray job submission (jobstepping).

I.2 \rightarrow{INTR} - Introduction to the documentation

This document is a complete description of the capabilities of the CCM Processor, including all of the information needed to run the code on the Crays at NCAR. A few 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 the NCAR Tech Note TN-290+IA "CCM Modular Processor Users' Guide (Version PROC02)" , October 1987.

This Users' Guide is designed primarily for reference; it is NOT intended to be read sequentially from beginning to end. New users are advised to read "Introduction to the UNICOS CCM Modular Processor", NCAR/TN-383+IA, December 1992, by L. Buja. This introductory document provides an overview of what the Processor can do, and includes a number of examples of simple processing runs. The next step is to become familiar with the organization and content of the complete Users' Guide. It is very important to read this introductory section fully before attempting to use the documentation; it is also recommended that the following section "I-3 Using the Processor" should be skimmed before running any important processing jobs.

In addition to the hardcopy NCAR Tech Note printing, this Users' Guide is also maintained on the SCD front-end machine at NCAR in a form suitable for online access. The online version is nearly identical in format to the printed version, but it's easier to use for quick referencing since an interactive utility has been provided to facilitate cross-referencing. To access the help software, log on to the front-end computer at NCAR, and issue the following command:

```
/crestone/u0/ccmproc2/doc/icp help
```

This will show how to use the standard icp facility to scan this document and look up help on individual topics. icp provides summaries and documentation of Processor ICPs. The command "icp" target displays the Processor documentation on the ICP target (i.e. "icp daysa" will display the explanation of the ICP DAYS A). Note that numeric substitution for the field pass number "n" is limited to the number "1". If an ICP matching target is not found, an icp -k target is automatically attempted. An "icp -k target" looks up information on the general category of the word(s) target in a summary of the ICP definitions.
Finally, an "icp -s target" searches through the entire online Processor documentation for "target".

Remote users can access icp online from their own UNIX platform via the remote shell command "rsh" to the SCD front end machine meeker.ucar.edu. The command would be of the form:

```
rsh meeker.ucar.edu /crestone/uO/ccmproc2/doc/icp target
```

where target is the input to the icp command (i.e. DAYSA). To run rsh on meeker, the remote user needs an account on meeker.ucar.edu and a valid .rhost file on the user's meeker.ucar.edu account (otherwise an error message of "permission denied" will result). Local CGD users can issue the command simply by entering "icp target" on the CGD system.

A printout of the icp interface to the emacs editor can be obtained via the command:
```
rsh meeker.ucar.edu cat /crestone/uO/ccmproc2/doc/icp.el
```

Since the Processor can perform a large number of complex processing tasks, a large and complex Users' Guide is required for complete documentation. The Processor has been designed, however, so that relatively simple tasks require a relatively low level of Processor familiarity. For example, if all analysis is to be performed on Model hybrid surfaces, then it is not necessary to become familiar with the vertical interpolation options. Wherever possible, options which do not need to be related are, in fact, completely independent in terms of both the code and the user interface. This philosophy has also been applied to the design of the documentation. In most cases the descriptions of specific capabilities stand alone; where background information is needed or there is potential interaction between options, cross-references are provided. The documentation has another feature intended to make it easier for experienced users to quickly locate frequently needed information: the general discussion of capabilities is separate from the information needed to actually request a particular option. The idea is that the general descriptions are relatively easy to remember and require only infrequent use, while specific usage information is referenced more often and needs to be succinct. These features add up to a somewhat unconventional approach to documentation, so it is very important that the basic organization and cross-referencing syntax be fully understood before attempting to use this Guide. The remainder of this introduction presents this information.

The general discussions of Processor capabilities are called either "Topics" or "Topic Discussions". These Topic Discussions briefly describe all of the options related to each capability, and cross-reference the detailed descriptions of the Input Control Parameters (ICPs) used to set the options. (The ICPs are simply predefined keywords which may be assigned one or more values on free-format input records set up by the user. This mechanism is described in more detail in separate Topics.)

When ICPs are referenced in the Topic Discussions, they are categorized as either "primary" or "secondary". Primary ICPs are used to control major processing characteristics such as whether or not a particular Module is to be executed; familiarity with all of a Topic's primary ICPs is recommended before the capability being described is used. Secondary ICPs control relatively...
minor aspects and usually default to the simplest or most commonly used values. Familiarity with these ICPs is not always necessary. The distinction between primary and secondary keywords is relative only to the referencing Topic; a primary ICP for one Topic may be secondary for another. The ICPs are categorized in this way solely for the purpose of making a general recommendation concerning familiarity with related options; the distinction in no way reflects an inherent attribute for a particular ICP, and does not imply which ICPs should or should not be specified when a particular capability is used. In other words, this is simply a documentation aid designed to help minimize the amount of material read by new or casual Processor users.

This documentation uses a specific syntax for cross-referencing both Topic Discussions and ICPs. Topic Discussion references consist of a four-character key, prefixed with a "*" and enclosed in parentheses. For example, (*INTR) is a reference to this Topic. References to ICP keywords consist of the keyword enclosed in brackets: square brackets are used for primary keywords while pointed brackets are used for secondary keywords. For example, [TAPESc] is a reference to the primary ICP TAPESc, while <TYPEc> is a reference to the secondary ICP TYPEc.

For the purpose of documentation only, ICP keywords are symbolically compressed if (and only if) they are "Case" or "field pass" dependent (i.e., a separate keyword exists for each possible Case and/or field pass; see (*DEFS) for a definition of these terms). When a referenced ICP keyword has a lowercase "c" suffix, it indicates that "A", "B", or "C" should be used in its place; 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 ICPs may not be defined for all possible Cases or field passes; exceptions are noted in the complete ICP descriptions.

This documentation is divided into two parts: I) descriptions of the processing capabilities (Topic Discussions), and II) an alphabetical listing of Input Control Parameter (ICP) descriptions. Each of these parts is described separately in the following paragraphs.

Part I contains a Topic Discussion for every major Processor capability. The first line in each Topic includes a 4-character Topic key preceded by the two characters "->". This key is used only for cross-referencing as described above, and is included in the Table of Contents as well. Most Topics discuss capabilities individually, and so do not need to be read sequentially in the order in which they appear. Where additional information is required in order to properly use a particular capability, cross-references are given. This part of the documentation is intended to be used as follows. Read or skim the text for a particular Topic with the following questions in mind: 1) Is this the appropriate Topic for the desired operation? If not, a cross-reference may lead to the correct Topic; 2) Are related capabilities also required, and if so, which ones? The text and cross-references should answer these questions; and 3) How are the appropriate options requested? At the beginning of each Topic is a list of all related ICPs, divided into the previously described primary and secondary categories. Following these ICP references into Part II should provide all the information needed to request a particular option. There is an additional, very important piece of information not generally provided by the Topic describing a capability: when, relative to other requested operations, is
a particular operation performed? To answer this question it is usually necessary to consult the order tables presented in the special Topic (*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.

Part II shows how to request the various processing options by listing the Input Control Parameter (ICP) keywords and the values that may be assigned to them on input records. The ICP descriptions include the associated keyword (it may be symbolically compressed as previously described), the minimum and maximum number of values which may be assigned, and the FORTRAN data type(s) of the value(s). Examples of option requests are also included for most ICPs. These descriptions are alphabetized by keyword.

All but the very simplest Processor jobs require a basic understanding of how the Processor works. The major section "Using the Processor" (*USPR) is intended to fill this need. It is highly recommended that at least the first five subtopics be skimmed by all users: (*DEFS), (*NICP), (*SICP), (*MSIN), and (*ERRS). The importance of occasionally referring to the ordering topic (*ORDR) has already been mentioned. Much of the information contained in the "Special Topics" section is related to running costs and efficiency, but there are also a few subsections describing special features.

This Users' Guide will be periodically updated, but reprinting will probably be less frequent than updating of the online version.

I.2.1. *DEFS - Definitions of Processor Terms and Acronyms

Definitions of some important Processor terms and acronyms are given below, in alphabetical order.

>blocked point - a gridpoint at which the value of some field (see definition below) 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 input fields grouped in order to allow comparison with another Case. Cases allow for independent data streams from which comparisons or merges can be carried out. Additionally, this allows for two separate processing steps in the same run. The primary (control) Case is designated "Case A", and the secondary (experiment) Case is designated "Case B". These two Cases may be quantitatively compared, and the result of such a comparison is designated "Case C" (*COMP). Alternatively, Case C may be created by merging Cases A and B (*CMRG).
If neither Case comparison nor merging is requested, the data being processed should be designated Case A. Some ICPs refer only to a specific Case, and are therefore characterized as being "Case-dependent". An ICP is Case dependent if, and only if, the last alpha character of the associated keyword is an "A", "B", or "C". (The last alpha character may be followed by a numeric digit designating the "field pass" number. See the definition of "field pass" below.) A Processor Case normally consists of a single time series in which each time sample has a unique "day" value (see definition of "day" below). A common example is a series of history tapes output by a single Model run. It is possible, however, for a single Processor Case to consist of more than one time series, e.g., a number of sets of history tapes output by a number of Model runs. This is referred to as "ensemble" processing (see definition below). The word "case" is sometimes used in this context with a variety of meanings, but such usage is avoided in this documentation. The first letter of the word "Case" is always capitalized if, and only if, Processor comparison Case A, B, or C is the intended meaning. Unless ensemble processing has been requested (*ENSB), all day values within the Case should be unique.

>CCM - the NCAR Community Climate Model. The CCM2 is the most recent version, CCM1 was the previous version and CCMOB is its predecessor. The CCM2 is maintained by the CCM Core Group (an administrative Group within the Climate Modeling Section of the Climate and Global Dynamics Division at NCAR). CCMOA is an older Model which predates the formation of the CCM Core Group.

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

>ensemble - a collection of time series data which are to be processed together as a single Processor comparison Case (see definition of Case above). For example, the same series of "days" may be output by a number of Model runs, and an average across Model runs may be produced. This is referred to as ensemble processing (*ENSB), and each time series (Model run) is referred to as a "member" of the ensemble.

>field - a set of values of some particular quantity defined on a set of points in time and three-dimensional space. A field may either initially exist on the input data tape or be derivable from other input 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 vary from one Model version to another, Model output field names are not listed in this documentation; it is usually necessary to consult the Users' Guide for the appropriate Model. 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; this is a fatal error.

>field pass - one pass through all input data and the associated processing. The number of fields which can be processed 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). Multiple field passes may be made for each Case (see definition above) being processed. Some ICPs refer only to a specific field pass, and are therefore characterized as being "field pass dependent". An ICP is field pass dependent if, and only if, the associated keyword ends in a numeric digit. This digit identifies the field pass number. Note: The field pass capability was created when memory was very limited and is rarely used anymore.

>History tape - a MSS (NCAR's Mass Storage System) data file, usually output by the Model, consisting of a sequence of logical records containing the values of Model fields 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 "files" on the MSS.

>Input Control Parameter (ICP) - a parameter which may be set by the user in order to control processing operations. Each ICP has an associated keyword which is assigned values on a set of statements in a special file contained within the Cray job deck. Strictly speaking, there is a logical distinction between the ICP and its associated keyword: the keyword is a symbolic name used to identify a particular ICP only for the purpose of assigning values to the ICP. From the user's point of view, however, the distinction is almost never important and the terms "ICP" and "ICP keyword" can be used interchangeably. The only exception occurs for the few ICPs which are associated with two, synonymous keywords; in these instances either keyword may be used to assign values to a particular ICP.

>jobstep - a single execution of the Processor code. A Processor job consists of one or more jobsteps. (*CJDK)

>latitude - latitude is defined as increasing northward, positive in the northern hemisphere, negative in the southern. Values are always specified in degrees. Latitude lists for CCM datasets are not regularly spaced, rather, they are arranged in a gaussian distribution.

>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 ident-
ifying a hybrid level (which is indicated by an 'H' suffix), is 1000.0 times the actual sigma value. The value identifying a pressure level (which is indicated by a 'P' suffix), has the units of millibars. The value identifying a potential temperature level (which is indicated by a 'T' suffix), has the units of degrees Kelvin.

>longitude - longitude is defined as increasing eastward, positive east of Greenwich, negative west of Greenwich. Values are usually in degrees between -180.0 and the last grid point before (west of) +180.0. The dateline is usually designated as -180.0.

>Model - The NCAR Community Climate Model (CCM); more specifically, the version of the CCM being processed. See the definition of CCM.

>MSS - The NCAR Mass Storage System; this system handles all of the users' longterm storage needs for large datasets. Cray disk datasets may be purged at any time and must be shipped to the MSS for archival storage.

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

>Save Tape - a data file output by the Processor and saved so that additional processing can be done at a later time. Save Tapes are usually MSS files, but may also exist (temporarily) only on the Cray disks.
To use the Processor, the user runs a UNICOS script on the NCAR Cray. This script first copies the Processor Input Control Parameter (ICP) instructions into a file called "parms", then calls the executable version of the Processor. Once running, the Processor will look for the file "parms", check the contents for correct syntax, then carry out the instructions given in the file.

The basic structure of any Processor script will be of the form:

Line | Script
----|----------------------------------
1    | `#!/bin/csh`
2    | `#QSUB -q econ -eo`
3    | `cd TMPDIR`
4    | `cat >! parms << 'END'
5    | `PROCESSOR INPUT CONTROL PARAMETERS (ICPs)
6    | `END'`
7    | `/ccm/pror/Processor`
8    | `exit`

where:

Lines 1-2 are Cray UNICOS shell commands controlling job execution in the batch queues (See the SCD's NCAR UNICOS Primer for more details).

Line 3 ensures that the Processor is run in an empty, temporary directory.

Line 5 copies the Processor's ICPs into the file "parms".

Line 6 represents a group of Processor ICP keywords that determine which processing tasks are carried out. The ICPs are all the instructions following the "cat >! parms << 'END'" line and preceding the next 'END' in the run script. See the Processor script, plot.sh, below for an example of ICP usage.

Line 7 is the target for the cat command in line 5. When the cat command encounters the 'END', it stops placing text into the file "parms".

Line 9 runs the Processor. The Processor will first check the ICPs for correct syntax, then carry out the instructions defined by the ICPs.

Line 10 exits the script.

The following sample Processor script, plot.sh, provides an example of how ICPs can be combined together to do the following:

- input days 11.0 through 15.0 from the history tape /CSM/ccm2/414/hist/h0002,
- for these days, read the temperature (T) field, which is on model levels,
- interpolate T from the model levels to the specified pressure levels,
- produce horizontal plots of T on the specified pressure levels,
- dispose the plot file to MP, the SCD laser printer.

The output job log and a sample plot for plot.sh are shown in Figures 6 and 7 on pages 54-57 in "Introduction to the UNICOS CCM Processor". Note: The actual script that is run does NOT contain the line numbers 1-21 under "Line". It will only contain the text under the heading "Script".

<table>
<thead>
<tr>
<th>Line</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>#!/bin/csh</td>
</tr>
<tr>
<td>2</td>
<td>#QSUB -q econ -lt 30 -IT 33 -lm 3Mw -lM 3Mw -eo</td>
</tr>
<tr>
<td>3</td>
<td>cd $TMPDIR</td>
</tr>
<tr>
<td>4</td>
<td>cat &gt;! parms &lt;&lt; 'END'</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
</tr>
<tr>
<td>6</td>
<td>C plot.sh read temperature from a history tape and plot</td>
</tr>
<tr>
<td>7</td>
<td>C</td>
</tr>
<tr>
<td>8</td>
<td>TITLEA = 'Plot.sh: 850,500,200mb T from the CCM2 Control Run'</td>
</tr>
<tr>
<td>9</td>
<td>TAPESA = '/CSM/ccm2/414/hist/h0002'</td>
</tr>
<tr>
<td>11</td>
<td>FIELDA = 'T'</td>
</tr>
<tr>
<td>12</td>
<td>PRESSLE = 850.,500.,200.</td>
</tr>
<tr>
<td>13</td>
<td>HPROJ = 'RECT'</td>
</tr>
<tr>
<td>14</td>
<td>HPCINT = 'T',850.,5., 'T',500.,10., 'T',200.,0.</td>
</tr>
<tr>
<td>15</td>
<td>DPLTMF = 'MP'</td>
</tr>
<tr>
<td>16</td>
<td>ENDOFDATA ---------------------------------</td>
</tr>
<tr>
<td>17</td>
<td>END</td>
</tr>
<tr>
<td>18</td>
<td>END</td>
</tr>
<tr>
<td>19</td>
<td>END</td>
</tr>
<tr>
<td>20</td>
<td>/ccm/proc/Processor</td>
</tr>
<tr>
<td>21</td>
<td>exit</td>
</tr>
</tbody>
</table>

Line-by-line explanation:

1. #!/bin/csh instructs the Cray UNICOS shell to run under the C-shell.

2. The #QSUB command supplies instructions to the batch queue controller. Here, the -q econ runs this job in the economy queue. Time limits are 30 seconds for each process (-lt 30) and 33 seconds for the entire job (-IT 33). Memory limits are 3 Megawords for each process (-lm 3Mw), as well as for the entire job (-lM 3Mw). The -eo combines both the diagnostic and the printed output into a single output file.

3. cd $TMPDIR makes $TMPDIR the current directory that the job will be run from. It is important that the Processor is run from an empty directory such as $TMPDIR, since extra files left in the working directory from previous runs can cause conflicts.

4. Blank lines are allowed in the UNICOS portion of the script.
5. cat >! parms << 'END' places all the text after line 5 until the line
'END' is encountered (on line 18), into the file "parms". The Processor
will then read the ICPs from "parms".

7. Comment lines may be used and are encouraged. Comment lines must start
with a either a lower or upper case 'C' in column one.

9. TITLEA = 'Plot.sh: 850,500,200mb T from the CCM2 Control Run' is a user
supplied title which will appear on all plots and printout.

10. TAPESA = '/CSM/ccm2/414/hist/h0002' is the full Mass Store filename for
the input CCM2 history tape that contains the temperature data.

11. DAYS = 11.,12.,13.,14.,15. specifies that these model time samples on
the history tape /CSM/ccm2/414/hist/h0002 will be processed.

12. FIELDA1 = 'T' is the list of names of fields to be processed. In this
case it is temperature. For a list of available fields for CCM1 and CCM2,
See section 3 of "Introduction to the UNICOS CCM Processor".

13. PRESSLE = 850.,500.,200. is the list of pressure levels, in millibars,
for vertical interpolation. If this ICP is not present, the default is no
interpolation from the input model levels.

14. HPROJ = 'RECT' specifies that all requested fields are to be plotted and
that an equatorial cylindrical equidistant projection will be used for all
horizontal plots.

15. HPCINT = 'T',850.,5.,'T',500.,10.,'T',200.,0. is an array of triplets
used to specify the contour intervals on horizontal projection plots. In
this case, T at the 850 mb level will be plotted with a contour interval
of 5.0K, T at the 500 mb level with a contour interval of 10.0K, and at
the 200 mb level, an appropriate value will be chosen automatically.
There will be a total of 15 plots produced; 3 plots per time period (one
for each level) for each of five time periods (days 11.,12.,13.,14.,15.).

16. DPLTMF = 'MP' specifies that the plot file is to be disposed to the SCD
high speed laser printer. The plots may be picked up in room 9A for
on-site users. Otherwise it will be mailed back to the user's remote site
within 3-6 days.

17. ENDOFDATA indicates the end of this Processor jobstep.

18. 'END' is the target for the cat >! parms << 'END' command.

20. /ccm/proc/Processor runs the Processor. Job accounting will be run and
printed at the end of the last jobstep.

21. exit ends the script. Unused ICP's may be stored after this line.

From the above example, it is clear that there are specific syntax rules for
the use of ICPs (see section 1.3.3 for a complete description). Some of the
most important are:

1. The general format is: ICP = VALUE(S).

2. The ICPs TAPESA, DAYSA, and FIELDA1 are all required for ANY processing to
be done. The one exception is that DAYSA is not used when a Time Averaged
Save Tape is input.

3. The first ICP line after the cat command must not be blank.

4. Start all new keywords in column 2 or farther right. ICPs may not extend
beyond column 72. Field names must be 8 characters or less.

5. ICPs may be continued to subsequent lines by a trailing comma on the
preceding line or a leading comma on the continuation line.

6. The letter C in column one indicates a comment line.

7. The TYPE of each value (i.e., INTEGER, REAL, or CHARACTER) is important.
   Follow FORTRAN syntax conventions for the value specifications.

8. Do not use tabs or trailing blanks.

9. All ICPs must be capitalized and all arguments are case sensitive.

10. ICP definitions may span multiple lines.

11. ICPs are order-independent, and each keyword can only be used once in
    each jobstep.

12. Each group of ICPs ends with an ENDOFDATA ICP.

Another item to mention is that some ICPs are required as inputs into the
script, while others can be left unspecified, in which case they are automatic-
ally set to a default value. Those keywords that are required have no default
value, while all other keywords have default values assigned to them within the
Processor code.

The Processor automatically transfers data files from the Mass Store to the
Cray disks using the filename specification from the TAPESA ICP. In this
example, the file /CSM/ccm2/414/hist/h0002 is read from the Mass Store System.
The DAYSA ICP (required for history tapes) specifies which model times are to
be processed and can either be requested explicitly (e.g., 11.0, 12.0, 13.0,
14.0, 15.0), or in shorter, do-loop notation (11.0, 15.0, 1.0). The FIELDA1
ICP (required) specifies the fields to process for this dataset; in this case
it is temperature T.

Other ICPs specify how the data will be processed. In the previous example,
HPROJ = 'RECT' (default is 'NO') specifies that horizontal projection plots for
all levels are plotted on a rectangular cylindrical equidistant projection.
The TITLEA ICP contains the user defined title for plots and printout of Case A
processing. The default is to produce no title.
Some ICPs end in an A, B, or C. Any ICP with such an ending is a case dependent ICP. For comparison of two cases (or tapes), Case A and B suffixes are used for ICPs pertaining to these two tapes, and the Case C suffix applies to the keywords pertaining either to the comparison of Cases A and B or to the merging of Cases A and B. In the no comparison case, Case A is used for all case-dependent keywords. The DAYSA ICP indicates the days for Case A that should be processed.

Those ICPs not ending in a capital A, B, or C are case independent ICPs and their instructions apply to all cases that are being computed. An example of a case independent ICP is PRESSLE.

Other ICPs, such as FIELDA1, also have a numeric suffix. The numeric ending indicates a field pass dependent ICP. Field passes allow the user to process input data over several field 'passes' of the input data, so that enough memory is available for each pass of the data. The FIELDA1 ICP is also an example of a field and case dependent ICP. Memory restrictions limit the number of fields that can be processed in a single pass. ICPs with no such suffix control processing for all field passes and cases.

In cases where the Processor cannot perform the desired processing task in one run, capabilities exist for running multiple Processor runs (or steps) within a single job submission. This is done simply by concatenating the different ICP scripts together, separated by ENDOFDATA ICPs. The Processor is executed repeatedly, once for each ENDOFDATA in the processor script (in the plot.sh example, only one jobstep is run). The major reason for jobstepping is in cases when the user needs a different order of operations. This capability, called jobstepping, often saves time submitting jobs that must be processed consecutively.

For example, suppose the job to be performed is a composite of January averages. Without jobstepping, the composite averaging job would have to be submitted after all the single jobs were completed. But by jobstepping, each step could compute one January average, while the last step could compute the composite average. This technique can also be used to avert unnecessary interaction with the Mass Store (by saving the results from one run on the CRAY disk and immediately reaccessing them, rather than disposing to the Mass Store). Note the provision in the ICP jobstream for jobstepping, by separating these two groups of control parameters into two separate Processor 'jobs' within the same job script (each execution of the Processor then uses the next script of ICPs). See example 10 on page 24 of 'Introduction to the UNICOS CCM Processor' for an example of a multi-step Processor job.
I.3.1.2 SAMP - Accessing the sample Processor job scripts

The easiest way to get started is to copy one of the sample Processor scripts, run it to make sure the job submission and retrieval method works, then modify the sample script for the particular application. A set of sample Processor codes are maintained at NCAR in the directory /ccm/proc/samples:

1. plot.sh  
   read and plot temperature from a CCM2 history tape.
2. initial.sh  
   make a CCM2 initial dataset.
3. one-day.sh  
   process one day for one case.
4. 30-day-ave.sh  
   produce a thirty day time average.
5. compare-aves.sh  
   produce two ten-day time averages and compare them.
6. rd-savetape.sh  
   recompute differences and plots from two save tapes.
7. plt-diffs.sh  
   replot only differences from a save tape.
8. hsl.sh  
   Read a history tape, output T to a Horizontal Slice Save tape, read the HSL tape with FORTRAN program.
9. std-dev.sh  
   compute standard deviations from four save tapes.
10. ave-ave.sh  
   produce two five-day time averages and average them.
11. derfld.sh  
   Make a simple derived field and plot it.
12. spectral.sh  
   Spectrally truncate a T42 dataset.
13. sample1.sh  
   Sample 1 plot example from Appendix A.
14. sample14.sh  
   Sample 14 plot example from Appendix A.

The first 12 sample codes are described in the document "Introduction to the UNICOS CCM Processor", NCAR/TN-383 STR. The remaining 14 codes are listed in Appendix A of this document.

There are a number of ways to get copies of these sample codes:

1. Remote users can issue the following command from their local UNIX machine to print out the sample codes:
   
rsh shavano.ucar.edu /ccm/proc/samples/list

   To copy individual sample files, such as plot.sh to the local directory, use the UNIX command:

   rcp shavano.ucar.edu:/ccm/proc/samples/plot.sh plot.sh

   (Note: An error message of "permission denied" probably means that the user's .rhosts file is configured incorrectly)

2. When logged onto shavano directly, the user can copy the sample codes to their own shavano directory by typing:

   cp /ccm/proc/samples/*.sh.

A number of additional sample ICPs for plotting data, complete with the resulting graphics, can be found in Appendix A of this document.
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 (*COMP) or merged (*CMRG), 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 in Part II. 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 produce a defined keyword.

The following is a summary of the syntactical rules for the input records that control Processor options. (Each line in the input file is a separate record.)

* The general format is ICP 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 specified values may vary within a range for many ICPs; the exact number provided 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 filter width.)

* 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, a fatal error results.

* The TYPE of values is significant. Follow standard FORTRAN conventions for the specification of INTEGER or REAL constants. Specifically, a decimal point implies REAL, absence of a decimal point implies INTEGER. Exponential form may also be used with REAL constants (e.g., 1.E36). CHARACTER values are enclosed by single quotes (apostrophes). All CHARACTER values are padded on the right with blanks so that the total number of characters is a multiple
of eight. In most cases an incorrectly specified type is recognized and results in an error message and an abort. For some ICPs that require mixed mode values, however, such an error is not trapped and the results are unpredictable.

* Continuation records are indicated by either a leading comma on the continuation record or a trailing comma on the previous record. Also, if a record contains only a keyword and equals sign, the following record is interpreted as a continuation. The keyword should not be repeated on continuation records.

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

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

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

* Repetition of a value may be indicated by prefixing the value with an integer repetition count, separated from the value with a colon. For example, 3:0.0 is equivalent to three zeros separated by commas.

* A complete set of records for a Processor run (jobstep) should be terminated with a record containing only the word ENDOFDATA. Any record beginning with the character string END has the same effect. Input records following the ENDOFDATA for the last requested jobstep are ignored.

* For multiple runs within a job (jobstepping) (*DEFS), each run should have its own, complete set of Input Control Parameter records. The first record of each set should immediately follow the ENDOFDATA in the previous set; these sets should NOT be separated by end-of-file delimiters.

I.3.4. ->MSIN - Mass storage interactions

Primary ICPs: none

Secondary ICPs: <MSPFXI> <MSPFXIc> <MSPFXO> <MSPFXOc> <MSTXTI> <MSTXTIc> <MSTXTO> <MSTXTOc> <MSDIRI> <MSDIRIC> <MSDIRO> <MSDIROC> <MSRTO> <MSRTOC> <SCRUB> <DELRLE>

The following discussion assumes that the reader is familiar with the basic terminology associated with the use of the Mass Storage System (MSS). A good understanding of these concepts is necessary in order to avoid naming problems that can result in unknowingly processing the wrong dataset. This section provides the details of the Processor's handling of Cray disk datasets, including the effect of the controlling ICPs.
Normally, all movement of data between the MSS and the Cray disks is handled internally by the Processor code and is transparent to the user. When execution reaches the point at which a particular MSS file is to be read, there is a check to see if the corresponding permanent dataset is currently accessed by the job. If not, msread is called, initiating a copy of the file from the MSS. By default, the Processor checks for the existence of the file on the Cray disks under the name /usr/tmp/ccm/MSSNAME where MSSNAME is the name of the MSS file being read.

The MSS pathnames (MSPNs) for both input and output data (*.DATA) are specified with ICPs; the ICP should contain, as CHARACTER data, either a "full" or "relative" MSPN. The input MSPN is considered to be "full" if, and only if, it begins with a slash; otherwise the MSPN is considered to be "relative" in the sense that at least the "home directory" (and possibly one or more subdirectories) has been omitted. If a relative MSPN is input, the Processor converts it to a full MSPN in a two-step process: 1) if an MSPN "prefix" has been specified (see ICP descriptions below), the prefix and relative MSPN are concatenated; 2) if step 1 results in a relative MSPN, a home directory is added, using the user's login name. The resulting full MSPN must be less than 80 characters, or a fatal error occurs.

NOTE: Strictly speaking, the character strings specified as values for the various MSPN ICPs do not need to be valid "pathnames" until after prefix concatenation; in other words, the break between the two parts may be at any character, including in the middle of a directory name or filename.

Both the CCM2 and the Processor have adopted the convention of keeping the temporary disk copies of data files being input and output to Mass Store (MSS) files under the disk directory: /usr/tmp/ccm. Any MSS files input or output by the Processor will exist on disk by the name: /usr/tmp/ccm/MSSPATH/Filename, where /MSSPATH/Filename is the mass store path name of the MSS volume being read or written. To keep the file from being scrubbed off the disk during the course of a run, the Processor establishes a link between a file in the local directory (usually $TMPDIR) and the copy of the MSS file under the /usr/tmp/ccm subdirectory. The ICP SCRUB can be used to force a dataset to be read directly from the MSS, ignoring the disk copy of the MSS file.

For instance, when the user requests the MSS file /CSM/ccm2/414/hist/h0002, the Processor will first check on the Cray disk for the existence of the file:

/usr/tmp/ccm/CSM/ccm2/414/hist/h0002.

If it exists, the Processor will link it to a local filename. If the disk file doesn't exist, the Processor will first create a disk subdirectory named /usr/tmp/ccm/CSM/ccm2/414/hist/, then the MSS file /CSM/ccm2/414/hist/h0002 would be read into the local directory (usually $TMPDIR) via msread. The local file will then be linked to the disk file:

/usr/tmp/ccm/CSM/ccm2/414/hist/h0002.

In either case, the Processor will actually read the data from the local (usually $TMPDIR) version of the file.
This method has a couple of advantages. First, only one copy of each MSS file will ever exist on the Cray disks. Second, the file in /usr/tmp/ccm remains on the Cray disk beyond the running time of the current job, cutting down on access to the Mass Store. Finally, if the user is running with $TMPDIR as their local directory, the linked copy of the file in the local directory won't be subject to the disk scrubber during the course of the run.

There are a couple of potential problems with this approach. First, subdirectories below /usr/tmp/ccm can be created without global read/write/execute permissions by a non-Processor job. In this case, anyone other than the owner trying to create subdirectories or read files into those subdirectories will get a permission denied error, which halts the Processor. When this happens, it is up to the user to contact the owner of the restricted directory about changing the permissions. This problem can be avoided through use of the command "umask 000". Second, it may be possible to overflow the disk space allotted to /usr/tmp/ccm. For this reason, the default for the ICP DELREL has been changed to 2, meaning disk copies of MSS files will be deleted as soon as possible.

After a successful return from msread, the MSS file 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.

The Processor also internally assigns a unique Dataset Name (DN) to each input or output dataset. The DN has the form "LDNnnnn" where "nnnn" is a sequence number, starting at 0001, assigned in the order in which the DNs are generated. These DN assignments are maintained across all Processor jobsteps in a given run. Although it is not normally necessary to know the assigned DNs, they can easily be determined for any particular Processor job by making an ICP "checkout" run (see comments in standard run deck for instructions). This type of run prints a table of DNs and full MSPNs for all MSS files referenced in the ICPs, then terminates before any files are msread.

In general, requesting files from the MSS one at a time minimizes the number of MSS stage-ups required. There are times, however, when this procedure can cause wasted CPU time and unnecessary stage-ups. If a large number of files 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 files before the Processor begins execution. The necessary msreads can be added to the Processor run script, or they can be submitted as a separate job. There is still a possibility, however, that the datasets will be purged from the Cray disks before the processing job terminates, especially if the job is run at low priority. There is also a
possibility that the Processor run will abort due to an incorrect deck setup, and the datasets purged before the job can be rerun. If an MSS file is read by adding UNICOS commands to the run deck, the convention of reading MSS files under the directory /usr/tmp/ccm must be used.

The Processor also has the capability of forcing a stage-up from the MSS for any or all files. The icp <SCRUB> forces all input MSS files to be directly read from the MSS. Disk copies of these datasets existing under the directory /usr/tmp/ccm are deleted.

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). This is not the case, however, if LSD processing is requested (*LSDP); LSD processing uses a separate input unit for each member of 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 member of the ensemble.

All output datasets are written to the MSS by the Processor code without waiting to make sure the write was successful. Except for graphics output datasets, copies of output datasets are maintained so that subsequent jobsteps should never require a stage-up in order to read them.

When a dataset is CLOSEd so that its associated unit number can be reassign- ed, it may also be removed from the Cray disks as well, depending on the value of the ICP <DELREL>. The advantage of removing the dataset is that the disk space is immediately freed, and so is available for a subsequent file stageup from the MSS (and, of course, the need for a later purge by the Cray disk scrubber is eliminated). The disadvantage is that if the file is needed again later, it will have to be staged up from the MSS.

Many of the ICPs relating to Mass Store and Cray disk interactions are Case-dependent, but also have Case-independent variations (*COMPl). If a Case- dependent ICP is used, it applies ONLY to mass storage interactions for that Case (A, B, or C). Case-independent ICPs are used for Case-independent interactions, and also for Case-dependent interactions when the appropriate Case-dependent ICP is not specified.

NOTE: All ICPs whose keywords end in A, B, or C (possibly followed by a numeric digit) are Case-dependent; all other ICPs are Case-independent.

Following is a brief summary of the ICPs relating to Mass Store and Cray disk interactions; see the ICP definitions in Part II for additional details and examples.

The following ICPs may be used to specify MSPN prefixes:

MSPFXI - Case-independent prefix for input datasets
MSPFXc - Case-dependent prefix for input datasets
I.3.5 ERRS

November 1993

MSPFXO - Case-independent prefix for output datasets
MSPFXOc - Case-dependent prefix for output datasets

For both input and output datasets, the following ICPs allow the specification of additional parameters in the TEXT field of the msread or mswrite:

MSTXTI - Case-independent string to be appended to TEXT field for MS msread
MSTXTIc - Case-dependent string to be appended to TEXT field for MS msread
MSTXTIO - Case-independent string to be appended to TEXT field for MS mswrite
MSTXTIOc - Case-dependent string to be appended to TEXT field for MS mswrite

The following ICPs allow the specification of directories on the Cray disks for copies of files being read from or written to the MSS:

MSDIRI - Case-independent ID for input permanent datasets
MSDIRIc - Case-dependent ID for input permanent datasets
MSDIRO - Case-independent ID for output permanent datasets
MSDIROc - Case-dependent ID for output permanent datasets

NOTE: If no directories are specified, then /usr/tmp/ccm will be used.

The following ICPs may be used to specify MSS retention times for output datasets:

MSRTO - Case-independent MSS retention time
MSRTOc - Case-dependent MSS retention time

There is also a special option related to MSS interactions; specifying a write password of 'NOMS' for any output Save Tape prevents that Save Tape from being written to the MSS; it is still SAVEd on the Cray disk as a permanent dataset.

I.3.5. ->ERRS - Error handling

Primary ICPs: [DEBUG]

Each successful execution of the Processor code (i.e., each jobstep) 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 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, or that the Processor actually accomplished what the user intended.

Most execution time errors are detected by the Processor code and result in: 1) an error message printed in the output file, 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. Following the traceback, a symbolic dump of all routines in the traceback, plus the common
blocks in these routines is carried out. Any error messages printed by the Processor or its utilities 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 log file should contain a system error message. The most common errors in this category involve problems associated with staging up MSS files. A common cause of this kind of error is an incorrectly specified MSS pathname; look carefully at the printout preceding the dump and check the pathname for the requested file. If it is not what was intended, check the ICP specifications (*MSIN). Additional printout of internal Processor functions may be generated with the DEBUG ICP.

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 MSS or Cray disk problems), but it can also be caused by an error in the Processor code (please save the dump!). When this error occurs in conjunction with derived field calculations, particularly when doing division or logarithms, the ICP <PRLIMR> can be used to search for the presence of zeros (division) or negatives (logarithms). 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.

Aborts due to insufficient memory allocation are also relatively common. In many instances, simply increasing the job's memory limits specified in the QSUB command remedies error aborts, particularly when ishell calls are involved.

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 misspelled, a keyword suffix has been used improperly, or a character string value is not enclosed in single quotes (*SICP). Syntax errors are sometimes not detected until the line following the one containing the error; this is often the case for extraneous or misplaced commas. After all ICP's have been read there is a check on the number and FORTRAN type of the values input. This is done by the main Processor code, and results in messages that are usually self-explanatory. Character 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. (Also 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). Incorrectly specifying the type of an input tape <TYPEc> can sometimes cause the Processor to make an unreasonable
memory allocation request, resulting in the insufficient memory error message. 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 internal error messages is produced by the Processor's Data Unit Manager, which is the Processor's database manager. These error messages are characterized by the printing of "Names" and "Numbers" of "Units" and "Records", and are almost always caused by problems in the code. Please report such errors with an e-mail message to ccmproc2@ncar.ucar.edu.

I.3.6. ->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) is the same for both the LSD and the standard drivers (*LSDP), but it does depend on whether gridpoint time averaging (*TAVG) and/or time filtering (*TFIL) have been requested. 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).

The sequences of operations for spectral processing (*ORSP), spatial averaging and related plotting (*ORSA), Case C processing (*ORPC), and Save Tape Output (*ORST) are outlined separately; these sequences are independent of time averaging and filtering.

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. 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 by 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; a "?" indicates that fields may or may not be changed, depending on qualifying options. The absence of these symbols indicates that the processing step has no effect on subsequent processing.

Although the order of operations is fixed by the Processor code, they can be effectively reordered by performing them in separate jobsteps (*DEFS) linked by Save Tapes. 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, spatial averaging, Case C processing, and Save Tape Output are implemented as separate sub-drivers, just as they are outlined here.
I.3.6.1 ORNN

begin time loop
   begin field pass loop
      begin AB Case loop
         + AB 1. read next day from input tape (*DATA)
         + AB 2. compute type 111, 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 sample to time series for plotting (*PTIM)
            AB 11. write this time sample to Save Tape(s) (*ORST)
      end AB Case loop
   +? C 12. create and process Case C (*ORPC)
      ABC 13. if last time in loop, plot all time series (*PTIM)
      ABC 14. print requested field values (*PRNT)
      end field pass loop
      end time loop
   ABC 15. dispose any Save Tapes not yet disposed

I.3.6.2 ORAN

begin field pass loop
   begin AB Case loop
   begin time loop
      + AB 1. read next day from input tape (*DATA)
      + AB 2. compute type 111, 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 sample to time series for plotting (*PTIM)
      AB 10. write this time sample to Save Tape(s) (*ORST)
      + AB 11. accumulate zonal eddy statistics (*ZEST)
      + AB 12. accumulate time average statistics (*TAVG)
end time loop

+? AB 13. spatial averaging and related plotting (*ORSA)
AB 14. write time average to Save Tape(s) (*ORST)

end AB Case loop

+? C 15. create and process Case C (*ORPC)
ABC 16. plot all time series (*PTIM)
and write Time Series Plot Save Tape (*OTSP)

ABC 17. print requested field values (*PRNT)

end field pass loop

ABC 18. dispose any Save Tapes not yet disposed

I.3.6.3. ->ORNF - Order for no 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 111, 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 end points of time series

$ 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 sample to Save Tape(s) (*ORST)
+? AB 13. spatial averaging and related plotting (*ORSA)
AB 14. print requested field values (*PRNT)

end time loop
end AB Case loop

AB 15. plot all time series (*PTIM)
and write Time Series Plot Save Tape (*OTSP)

end field pass loop

AB 16. dispose any Save Tapes not yet disposed

I-24
I.3.6.4. ->ORAF - Order for 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 111, 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 9-13 only done away from end points of time series

$ 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 sample to Save Tape(s) (*ORST)
+ AB 13. accumulate time average statistics (*TAVG)
+$ AB 14. spatial averaging and related plotting (*ORSA)
  AB 15. write time average to Save Tape(s) (*ORST)

end time loop
end AB Case loop

+? C 16. create and process Case C (*ORPC)
  AB 17. plot all time series (*PTIM)
    and write Time Series Plot Save Tape (*OTSP)
  ABC 18. print requested field values (*PRNT)

end field pass loop

ABC 19. dispose any Save Tapes not yet disposed

I.3.6.5. ->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 (*DFLD)
+ 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. output savetapes (*SPEC)
  AB 11. draw spectral graphics (*SPGR)

I-25
- AB 12. delete fields in requested delete list (*SPEC)
  AB 13. reorganize memory (*CMEM)
$ AB 14. spectral to gridpoint transform (*SPEC)
+ AB 15. compute type 24 derived fields (*DFLD)
+ AB 16. add fields in requested exclusion list (*SPEC)

I.3.6.6. ->ORSA - Order of spatial averaging and related plotting

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

I.3.6.7. ->ORPC - Order of Case C processing

+ C 1a. compute differences and/or ratios to form Case C (*COMP)
  or
  + C 1b. merge Cases A and B to form Case C (*CMRG)

? C 2. spatial averaging and related plotting (*ORSA)
  C 3. append this time to time series for plotting (*PTIM)
  C 4. write this time sample to Save Tape(s) (*ORST)

I.3.6.8. ->ORST - Order of Save Tape output

NOTE: Time averages are written only for the last time in loop

ABC 1. write time average to Time Average Save Tape (*OTAV)
ABC 2. write next time sample (or time average) to Time Series Save Tape (*OTSR)
ABC 3. write next time sample (or time average) to History Save Tape (*OHST)
ABC 4. write next time sample (or time average) to Horizontal Slice Save Tape (*OHOR)
ABC 5. write next time sample to List Sorted Data Save Tape (*OLSD)
1.3.7. 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 (*MLTM), processing costs (*COST) and conserving memory (*CMEM). The remaining topics deal with fields that 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).

1.3.7.1. 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, but it does occasionally present problems.

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] input records. 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. This problem is best approached by determining the number of "field-levels" that will fit into the "available memory". A "field-level" is one level of one field; for example, an eighteen-layer Model defines temperature at eighteen levels in the free atmosphere, plus the surface, for a total of nineteen field-levels. Memory used to store the field values is managed dynamically by the Processor. The "available memory" for this purpose is the maximum number of words which can be allocated to one job, minus the Processor's load field length (which is just the length of the code itself), minus the size of the input buffer(s) and other miscellaneous memory needs. On the Cray YMP-48, the "available memory" ranges from about 2 million words up to 16 million words for normal jobs that don't invoke special job classes. For a T42 dataset with 64 latitudes and 130 longitudes (including two wraparound points), there are 8,320 grid points on a horizontal surface. Allowing for some memory usage

I-27
inefficiency due to dynamic management, this implies that on the order of a thousand field-levels will fit into the available memory on the YMP. If time averaging is to be done, the number of field-levels which can be processed is cut in half, since space must be allocated for an accumulation of each field being averaged. Operations such as spectral processing (*SPEC) require additional space, and time filtering (*TFIL) with 31-point filters is considerably more restrictive since 31 time samples must reside in memory simultaneously. For these kinds of processing tasks, the most effective way to determine Processor limitations is trial and error. 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 that cannot be handled by the field pass approach.

I.3.7.2. ->COST - Processing costs

Primary ICPs: [FIELDcn] [DAYSc]

Secondary ICPs: <LYRSUBc> <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. One major exception is runs which produce a large number of horizontal projection contour plots. Since drawing a T42, global contour plot requires about 0.9 seconds of CPU time on the Cray Y-MP, 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 that 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 an estimate of plotting time. 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 if the absolute binary form of the Processor code is not used (*ARCH). 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 that can be used to reduce processing costs. The efficiencies of memory (*CMEM) and mass storage usage (*MSFH), 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 that 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 levels when a history tape is input; see (*LINL) for a discussion. Different levels of different fields can be excluded by defining new derived fields that are single, specific levels of existing fields, and by 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), often considerably reducing the amount of data that must be staged up from the MSS. The MSS data transfer cost is not included in the job's cost or indicated on the Cray output, but it is directly charged to the account under which the job was run.

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, however, that in order to interpolate time averaged data to pressure surfaces, surface pressure must be explicitly time averaged and saved (*VPRS).
Conserving memory

Primary ICPs: [FIELDcn] [MEMCON] [MEMORY]
Secondary ICPs: <LYRSUBc> <SURFLEV>

Optimizing memory usage for specific Processor tasks may make possible some operations which would otherwise require too much memory, and can significantly decrease the cost of processing. Conserving memory can lower costs directly because the charging algorithm is based, in part, on the average number of megawords of memory used by a job; 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] <LYRSUBc> <SURFLEV>. There are several additional memory conservation techniques that 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. 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, and memory would be wasted for those Modules not executed. 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 (deallocation) 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
consume 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 that 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 dif-
fferent 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 com-
pleted successfully many times. The fragmentation can be eliminated by writing 
the data to a temporary disk dataset, freeing all allocated blocks, then read-
ing the data back into memory. The [MEMCON] option performs this memory "reor-
ganization" 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 transfor-
mation of spectral data back into gridpoint space (where a few large blocks of 
memory are needed). 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 that are explicitly being excluded from spectral 
operations <SPCEFcn>. 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 some 
guidelines). 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, spec-
ifying 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. This kind of control is not possible when running from the absolute binary file, which was created using the loader defaults (*ARCH).

1.3.7.5. →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 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.

1.3.7.6. →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).
I.4 DATA

This section describes the input of Model history tapes (*IHST), and the input and output of Processor Save tapes (*SAVT). Only the data manipulation aspects of this I/O are presented here; for a discussion of mass storage interactions and naming conventions, see (*MSIN).

I.4.1. ->IHST - Input of History tapes

Primary ICPs:

[TAPESc] [TYPEc] [DAYSc] [FIELDcn] [DAYTYPc]
[DLDAYSc] [NINTAPc] [PRNTHD] [LEAPY]

Secondary ICPs:

<TITLEc> <ENSBLC> <LYRSUBc> <SURFLEV> <REORDER>
<DELREL> <SCRUB> <HYBASCc> <HYBPRBc>

History tapes produced by a number of different Models or previous Processor runs may be input to the Processor [TAPESc]. Acceptable Model versions include CCM2, CCM1, CCM0B and CCM0A (grid point format only). Processor History Save Tapes (*OHST) containing data on hybrid, sigma, pressure, or potential temperature surfaces may also be input [TAPESc]. In general, the type of history tape must be specified with [TYPEc], but this ICP defaults to a mode which automatically recognizes all of the history tape formats produced by CCM2, CCM1 and CCM0B. For all input history tapes, the packing density and time and space resolutions are all handled automatically.

Any subset of Model fields [FIELDcn] can be processed for any subset of available Model time samples [DAYSc] on any specified series of history tapes [TAPESc]. Multiple do-loop sequence of input days can be specified with [DLDAYSc]. [NINTAPc] may be used to automate the input tape list for sequential input tape numbers. The days requested with [DAYSc] may be interpreted as either absolute or relative to the beginning of the Model integration [DAYTYPc] (in most cases, however, there is no difference). Tape positioning is automatic both forward and backwards, including across tape boundaries. Input may also be limited to a subset (same for all fields) of available vertical surfaces (*LINL). <LYRSUBc> may be used to limit the vertical layers to be processed. Surface level fields (*XBTL) may be automatically associated and processed with the corresponding multilevel fields <SURFLEV>. The information contained on the input history tape header can be printed with [PRNTHD]. The ICPs <HYBASCc> and <HYBPRBc> allow the user to control the processing of hybrid coordinate history tapes.

[LEAPY] controls whether or not leap years are taken into account in all date computations <DAYTYPc>. A mismatch between this ICP and the Model's date computations may result in an inappropriate date request when DAYTYPc='DATE' and do-loop notation is used for DAYSc.

For most processing it is assumed that each requested time sample is unique within the entire set of input history tapes. Major exceptions occur for ensemble processing (*ENSB) and List Sorted Data processing (*LSDP), but even if one of these types of processing has not been requested, the uniqueness
I.4.1.1 LINL - Limiting history tape input levels

Primary ICPs: [LYRSUBc]
Secondary ICPs: <TYPEc>

When data are input from history tape(s), it is possible to limit processing to any subset of available free atmosphere levels (same set for all fields). The ICP [LYRSUBc] may be used to specify an arbitrary list of level indices for data on hybrid, sigma, pressure, or potential temperature surfaces <TYPEc>. This option is not available for input from sources other than history tapes.

If a field is defined at full hybrid levels (layer midpoints), the levels processed are those located at the midpoints of the requested layers (*DEFS). For fields defined on half-levels (layer interfaces), the levels processed are those at the bottom of the requested layers. For example, layer 1 for geopotential height Z 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 that 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 that 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 level subset. Note, that if the subset is vertically interpolated (*VERT), the results may 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 Z) is NOT considered to be an extra bottom level when counting levels for this ICP.

I.4.2 ENSB - 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 "ensemble
processing", and each Model run is referred to as a "member" of the ensemble.

When 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 belong to a
different member of the ensemble, and each requested day is processed from each
requested tape. Note that the use of a simple, one-dimensional list of input
tapes restricts ensemble processing to one input tape per member.

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

It is possible to define one ensemble average as comparison Case A <TAPESA>
<DAYS A>, another ensemble average as Case B <TAPESB> <DAYS B>, and compute the
difference between the two ensemble averages. Ensemble processing is imple-
mented in a way that minimizes the amount of memory needed, but this results in
inefficient dataset positioning. See (*MSFH) for a discussion.

The capabilities provided by this option are somewhat limited, and the
implementation is not particularly efficient for the kind of complex processing
often required for forecast studies. See (*LSDP) for an alternative which may
be more appropriate.

I.4.3 LSDP - List sorted data (LSD) processing

Primary ICPs: [DRVRTYP] [SAVLS1c] [RTLS1c] [PKLS1c]

Secondary ICPs: <NSBTAPc> <NSBDA Yc> <PWLS1c> <TITLS1c>

Much of the data organization within the Processor is designed primarily for
the tasks typically associated with the evaluation of a climate model. In
general, the processing needs associated with a forecast model are also met,
but the code design sometimes imposes limitations and inefficiencies. List
Sorted Data (LSD) processing provides a partial solution to these problems by
using an alternative Processor driver whose primary purpose is the output of a
specialized Save Tape designed to interface with other processing packages.
Although LSD processing provides for a more general way to input ensembles
(*DEFS), it does not provide for the computation of ensemble statistics, as
with ensemble processing (*ENSB). LSD processing does, however, allow the
processed data to be reordered (sorted), with respect to ensemble member, time,
and field/level as it is output. When run in LSD mode the Processor performs
three basic functions: 1) reads input data for some user-defined ensemble
having an arbitrary number of members; 2) selects a subset of time samples and
field/levels according to user-specified lists; 3) reorders (sorts) the data
and outputs them as horizontal slices to LSD Save Tapes (*LSDS). Note that the
data sorting affects ONLY the output LSD Save Tape; although other Save Tapes
may be simultaneously written (*SAVT), the ordering of the data on them is the
same as for standard processing.
LSD processing is requested by using [DRVRTYP] to specify that the LSD driver be used; also required is a list of output LSD Save Tapes [SAVLS1c], a record ordering code [RTLS1c], and a packing density [PKLS1c]. A descriptive title <TITLS1c> and MSS write password <PWLS1c> are optional, as are the ICPs <NSBTAPc> and <NSBDAYc>, which provide alternate specification of the input tapes <TAPESc> and days <DAYS1c> lists, respectively, in a generalized two-dimensional form appropriate for specifying ensembles.

The order of operations in the LSD driver is basically the same as in the standard driver (*ORDR), but not all standard Processor options are supported. Only field pass 1 may be processed, and Case comparisons (*COMP) are not recommended; other options may function properly, but most have not been tested at all. Thus, it is probably best to perform complex processing first using the standard driver, then use the LSD driver to write the LSD Save Tapes in a final jobstep. In addition, the spectral harmonic coefficients and related statistics may be output on LSD savetapes.

I.4.4. ->PRNT - Printout of field values

Primary ICPs: [PRINTc] [PRNTHD]

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 gridpoints <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. 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).

Since this printing is the last operation performed by the Processor (*ORDR), it reflects the results of virtually all previous processing operations. The only exceptions are options (such as certain forms of spatial averaging (*SPAV)) which affect the fields only temporarily. If the data have been averaged in one or more spatial dimensions, the location parameter(s) for the averaged dimension(s) is the four-character string "AVRG".

Header information from input history tapes may be printed with [PRNTHD].
There are currently seven different types of Processor Save tapes. Although each type is designed for a particular purpose, there is considerable overlap, and some purposes may be equally well served by more than one type. Following is a list and very brief description of the available types.

* Time Average (*STAV) - contains the single time sample resulting from a time averaging operation; intended for later Processor input; only form of Save Tape which can handle multiple field passes in a general way.

* Time Series (*STSR) - contains an arbitrary number of sequential time samples or a time average; intended for later Processor input.

* History (*OHST) - contains an arbitrary number of sequential time samples or a time average; intended for Processor, CCM, or user code input.

* Horizontal Slice (*OHOR) - contains horizontal slices of field values for an arbitrary number of sequential time samples or a time average; simplest form of Save Tape intended for user code input.

* List Sorted Data (*LSDS) - contains horizontal slices of field values and/or spectral harmonic coefficients and statistics for an arbitrary number of ensemble member and time samples; most general form of Save Tape intended for user code input.

* Surface Type (*SFCT) - contains time independent land, ocean, sea ice flag; intended for later Processor input for surface type masking.

* Time Series Plot (*STSP) - contains the reduced data for time series plots; intended for later Processor input for the purpose of replotting with different plot characteristics.

Each of these Save tapes is described in detail in the following topics. The remainder of this topic discusses a few common Save Tape features.

Any or all types of Save Tapes can be output in a single jobstep, including more than one type for a single Case. For input, Case A may read a different type than Case B, but different types cannot be mixed within a single Case.

If a field is being processed within the applicable Case(s) at the time a Save Tape is written, then that field is included on the Tape. If vertical interpolation (*VERT) is requested, then the data is written on the interpolated surfaces. Surface Type Save Tapes and Time Series Plot Save Tapes are Case-independent, but all other types of Save Tapes are Case-dependent (i.e., they contain data for only one Case). For most Processor jobs, all fields explicitly requested, and only those, are written to the Save Tapes. There are exceptions, however, and for some jobs it may be necessary to carefully examine the order of operations (*ORDR) to determine which fields are written to a particular Tape.

The Processor does not provide a general mechanism for automatically limit-
I.4.5.1 STAV

November 1993

I.4.5.1.1 ITAV

ing the size of the Save Tapes it produces. The number of time samples written to Time Series and History Save Tapes can be controlled, but it is up to the user to determine if the defaults need to be changed. As a general rule of thumb it is best to limit all output datasets to no more than about 150 megabytes of primary data (field values). The amount of descriptive information added to this varies according to the type of Save Tape, but it can be a very significant fraction if the amount of primary data is small. The largest Cray dataset that can currently be written at NCAR is 195 million bytes.

Only Horizontal Slice Save Tapes (*OHOR), History Save Tapes (*OHST) (*IHST), and List Sorted Data Save Tapes (*LSDS) are suitable for input to programs other than the Processor. The other forms of Save Tapes are designed to be used in conjunction with the Processor's own database manager, and are difficult to handle without it.

History (*OHST), Time Series (*STSR), and Horizontal Slice (*OHOR) Save Tapes are designed primarily for containing time series, but a time average (*TAVG) computed in the same jobstep that writes these tapes may be output instead. This "time average" mode is applicable only to time averages computed in the same jobstep in which the tapes are output. Time averages computed in previous jobsteps should be handled as a single time sample in "time series" output mode.

Only the data manipulation aspects of Save Tapes are presented in the following topics; for a discussion of mass storage interactions and naming conventions, see the special topic (*MSIN).

I.4.5.1. ->STAV - Time Average Save Tapes

Time Average Save Tapes contain a single time average, including the associated time average statistics (*TAVG), 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, but they are not suitable for input to other codes. Multiple field passes are handled automatically, and the field pass groupings may be different in the input and output runs.

Since Time Average Save Tapes contain data in the same format that is used internally in the Processor's database, all descriptive information remains associated with the data (including a processing history). The fields written to the Save Tape may also be vertically (*VERA), meridionally (*MERA), and/or zonally (*ZONA) averaged.

I.4.5.1.1. ->ITAV - Input of Time Average Save Tapes

Primary ICPs: [TAPESC] [TYPEc] [FIELDcn]

Secondary ICPs: <TITLEc> <DELREL> <SCRUB>

I-38
I.4.5.1.2 OTAV

One or more Time Average Save Tapes may be input and processed as a time series. Each time average is treated just like any unaveraged, single time sample. The [TAPESc] list therefore completely determines the time samples to be processed; the <DAYSc> list has no meaning and is ignored. A series of time average statistics may itself be time averaged, producing "second order" (or higher) time average statistics [TAPESc] [TYPEc] such as a standard deviation of monthly averages (*TAVG). Any subset of fields from the Save Tape may be processed [FIELDcn], and this subset may contain fields originally written in different field passes. For each requested field, all available vertical levels are always processed. The input data may be vertically interpolated (*VERT) only if it is defined on hybrid or sigma surfaces, and if surface pressure (PS) is explicitly requested [FIELDcn] and available.

If a series of Time Average Save Tapes is input for a single Case, then it is recommended that each tape in the series be produced by identical Processor runs or jobsteps. In general this is not necessary, but there are a few pathological cases that are not properly handled, particularly when time averaging (*TAVG) is requested.

When a Time Average Save Tape is input, a description of the contents is printed in the output file. This description, however, is based on the REQUESTED processing for the run that wrote the tape, not the actual processing performed. The actual processing is normally the same as the requested processing, but if an error was encountered in the run that wrote the tape, the tape description may not be correct. Even in this event, however, subsequent processing should proceed correctly since the erroneous description is used only in the printout.

I.4.5.1.2 -> 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 if, and only if, time averages are requested [TIMAVGc] (but a time series of length one can be averaged); each Case is requested separately and saved on a separate tape [SAVTAVc]. All field passes (*DEFS) are written to the same tape. All fields that are being processed at the time the tape is written (*ORDR) are output, including all time average statistics <CVFLDcn> <SDFLDcn> <TCFLDcn> <PRFLDcn> <ZSTFLcn> <ZCVFLcn>.

I.4.5.2 -> STSR - Time Series Save Tapes

Time Series Save Tapes contain a series of one or more time samples for some set of processed fields. The time series may span multiple tapes, but a series can contain only one field pass (*DEFS). These tapes may be both input (*ITSR) and output (*OTSR) by the Processor, but they are not suitable for input to
other processing programs. Time series Save tapes are especially useful when used in conjunction with time filtering (*TFIL).

Since Time Series Save Tapes contain data in the same format that is used internally in the Processor's database, all descriptive information remains associated with the data (including a processing history). The fields written to the Save Tape may also be vertically (*VERA), meridionally (*MERA), and/or zonally (*ZONA) averaged.

I.4.5.2.1. ->ITSR – Input of Time Series Save Tapes

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

Secondary ICPs: <TITLEc> <DELREL> <SCRUB>

A series of one or more Time Series Save Tapes may be input [TAPESc], and any subset of available time samples can be processed [DAYSc]. [NINTAPc] may be used to automate the input tape list for sequential input tape numbers. Tape positioning is automatic both forward and backwards, including across tape boundaries. Any subset of fields from the Save tape(s) may be processed [FIELDcn], but for each requested field, all available vertical levels are always processed. The input data may be vertically interpolated (*VERT) only if it is defined on sigma or hybrid surfaces, and if surface pressure (PS) is explicitly requested [FIELDcn] and available. Multiple do-loop sequence of input days can be specified with [DLDAYSc].

I.4.5.2.2. ->OTSR – Output of Time Series Save Tapes

Primary ICPs: [SAVTSRc] [NSVTSRc]

Secondary ICPs: <DAYSc> <NDYTSRc> <PWDTSRc> <SAVMTSR>

All fields being processed in field pass number 1 (and only those) are output. Normally all processed time samples are output, in the order in which they are requested <DAYSc>. If time average output mode <SAVMTSR> is requested, however, the time average computed in the same jobstep is written instead. Output for each Case is requested separately and saved on a separate series of tapes [SAVTSRc]. [NSVTSRc] may be used to automatically generate a large number of sequential output tape names.

If time averaging (*TAVG) or time filtering (*TFIL) is requested, then only the time average may be output for Case C, and then only if "time average" output mode is selected <SAVMTSR>. These restrictions are imposed by the fact that Case C does not exist inside the time loop when time averaging or time filtering has been requested (*ORDR). 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

I-40
series is specified by <NDYTSRc>, and it is the responsibility of the user to ensure that this ICP is set appropriately.

I.4.5.3. SHST - History Save Tapes

History Save Tapes contain a series of one or more time samples for some set of fields. The time series may span multiple tapes, but a series can contain only one field pass (*DEFS). These tapes may be both input (*IHST) and output (*OHST) by the Processor, and since they are in the same format as history tapes written by CCM2, CCM1 or CCMOB, they are suitable for input to these Models as well as other processing codes. See Appendix B for an example of how to produce a history tape suitable for starting a CCM2 run.

There are three slightly different history tape formats used by CCM2, CCM1 and CCMOB. The specific configuration for each of these are given in the Users’ Guides for each of the respective models. Since the Processor automatically distinguishes between the different data formats, there is no need to specify which CCM history tape format is being input to the Processor (*IHST). Since the Processor has an option to output any of the three formats (*OHST), it may also be used as a translator.

Although History Save Tapes written by the Processor can be input by the Model, there are a few minor differences in tape format which may cause problems, mostly in other programs designed to read Model history tapes; see (*OHST) for additional information.

Vertical (*VERA), meridional (*MERA), and zonal (*ZONE) averages may be written to a history tape with one exception: zonal averages cannot be written to the old, CCMOB single-record header format <OFTHSTc>.

I.4.5.3.1. OHST - Output of History Save Tapes

Primary ICPs: [SAVHSTc] [NSVHSTc]
Secondary ICPs: <NDYHSTc> <PWDHSTc> <PWHSTc> <PKHSTc> <SAVMHST> <OFTHSTc>

All fields being processed in field pass number 1 (and only those) are output. Normally all processed time samples are output, in the order in which they are requested <DAYSc>. If time average output mode <SAVMHST> is requested, however, the time average computed in the same jobstep is written instead. Output for each Case is requested separately and saved on a separate series of tapes [SAVHSTc]. [NSVHSTc] may be used to automatically generate a large number of sequentially numbered output History Save tape names.

Fields read from the input tape(s) are written in the order in which they are requested; other fields are appended in the order in which they are computed. The field ordering is summarized in the printed output. Specific MSS
Either the old, single-record header format or the new, three-record header format may be produced <OFTHSTc> (*SHST). The default is to output with the format as the input data. Two additional format conventions are also imposed by the Processor: 1) longitudinal wrap-around points are never written to three-record header format tapes, but two are always written to single-record header format tapes; and 2) the format type code on the header is modified. The units digit (decimal) remains unmodified (it is the only nonzero digit for tapes written by the Model), but the hundreds digit is always set to 1, and the tens digit is set according to the vertical coordinate on which the fields are defined: 0 for sigma, 1 for pressure, 3 for potential temperature, and 4 for hybrid.

If time averaging (*TAVG) or time filtering (*TFIL) is requested, then only the time average may be output for Case C, and then only if "time average" output mode is selected <SAVMHST>. These restrictions are imposed by the fact that Case C does not exist inside the time loop when time averaging or time filtering has been requested (*ORDR). 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>, and it is the responsibility of the user to ensure that this ICP is set appropriately.

The data may exist on any vertical surface at the time it is being written (*VERT). If the surface is hybrid or sigma, then the tape is identical in format to a CCM1 or CCMOB history tape, with the exceptions noted above. If the data have been interpolated to pressure or potential temperature surfaces, then the format is modified in four respects: 1) the header vertical coordinate array contains surface values ordered from bottom to top in either millibars or degrees Kelvin; 2) all vertical slices in the data records are ordered bottom to top; 3) the data records (latitudes) are ordered south to north; and 4) The tens digit of the header format type code is set as described above.

Data containing blocked points (*DEFS) cannot normally be written to History Save Tapes, since such data are not properly handled when input to the Model, and may not be properly handled when input to the Processor. (There is no check for blocked points when computing derived fields of types 11, 12, 13, and 132). By default, data are checked for blocked points as the tapes are written, and an error abort results if the special value (1.0E36) is encountered. The ICP <BPHSTc> may be used to override this check.

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>. History tapes intended for Model input should not be packed.
Horizontal Slice Save Tapes contain a series of one or more time samples for some set of processed fields. The purpose of these tapes is to provide processed data, in a simple format, to other processing programs. Horizontal Slice Save Tapes cannot be read by the Processor. By default, each record on the save tape contains one horizontal slice of field values; there is no information describing the data and these tapes cannot be processed further by the Processor. An optional header may be requested with the [SAVFHSL] ICP. All levels of all fields being processed in field pass 1 (and only those) are output (*DEFS). Also see LSD Save Tapes (*LSDS) for a more general way of writing horizontal slice data for input to other programs.

Primary ICPs: [SAVHSLc] [SAVFHSL]

Secondary ICPs: <SURFLEV> <LYRSUBc> <PRESSLE> <FIELDcn> <DAYSc> <ENSMBLc> <SHSLZAV> <SAVMHSL>

All fields being processed in field pass number 1 (and only those) are output. Normally all processed time samples are output, in the order in which they are requested <DAYSc>. If time average output mode <SAVMHSL> is requested, however, the time average computed in the same jobstep is written instead. Output for each Case is requested separately and saved on a separate series of tapes [SAVHSLc]. An informational header may be included [SAVFHSL].

The fields written to the Save Tape may also be vertically (*VERA), meridionally (*MERA), and/or zonally (*ZONA) averaged; the writing of zonal averages, however, should be requested with <SHSLZAV>.

Fields read from the input tape(s) are written in the order in which they are requested; other fields are appended in the order in which they are computed. The field ordering is summarized in the printed output.

If time averaging (*TAVG) or time filtering (*TFIL) is requested, then only the time average may be output for Case C, and then only if "time average" output mode is selected <SAVMHSL>. These restrictions are imposed by the fact that Case C does not exist inside the time loop when time averaging or time filtering has been requested (*ORDR). There are no such restrictions on Cases A and B. The output time series may be of arbitrary length, but it must be written to a single tape.

The data may exist on any vertical surface at the time it is written (*VERI).

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:
**SFCT**

**November 1993**

* Longitude varies first, starting at the dateline (180 degrees 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>. <LYRSUBc> may be used to limit the vertical layers to be processed. 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.

A HSL header is available via the icp <SAVFHSL>. If SAVFHSL = 'HEADER', each horizontal slice of data will be preceded by a seven word header giving:

1. Record Number. (integer)
2. Variable Name. (character*8)
3. Vertical Level. (character*8)
4. Number of longitude points. (integer)
5. Number of latitude points. (integer)
6. Day number. (real)
7. Name of this HSL tape. (character*80)

**SFCT**

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 conjunction with data that have the same horizontal resolution; input Save Tapes are checked for consistency with the data being processed.

Two T42 Surface Type Save Tapes, containing the sea ice distributions for January and July from the CCM2 414 control run, have been archived on the MSS: /CCMPROC2/ccm2/414/sfct/030101 and /CCMPROC2/ccm2/414/sfct/030701. The two Surface Type Save Tapes for R15 data are: /CCMPROC/SR15JA and /CCMPROC/SR15JL.

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.
I.4.5.5.1 ISFT

I.4.5.5.1. ->ISFT - Input of Surface Type Save Tapes

Primary ICPs:  [SFCTTAP]
Secondary ICPs: none

An input Surface Type Save Tape must be specified if, and only if, surface type masking is requested and the surface type information is not available from an appropriate history tape. See the pertinent masking topic for a more detailed discussion.

I.4.5.5.2 OSFT

I.4.5.5.2. ->OSFT - Output of Surface Type Save Tapes

Primary ICPs:  [SFCTCRTC]
Secondary ICPs: <TYPEc>

A Surface Type Save Tape can be created [SFCTCRTC] if, and only if, a history tape is used as input <TYPEc> for Case A or B. If both Cases A and B are input from history tapes, Case A is used as the source.

The following creates a T42 Surface Type Save Tape for January:

C Sample to create a T42 Surface Type Save Tape
C 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.
C
TITLEA = 'T42 January Surface Type Save Tape'
TAPESA = '/CSM/ccm2/414/hist/h0086'
DAYSA = 030101.
DAYTYPA = 'DATE'
FIELDA1 = 'SFCT'
DERFLD = 'SFCT',111,0,31,0
     , 'ORO',0.5,'GE', 'ORO',1.5,'LT',':AND'
     , 'ORO',0.5,'LT', -1.,':TIMES', ':PLUS','.END'
SFCTCRTC = '/USERNAME/ccm2/414/sfct/030101','passwd'
MSRTO = '365'
ENDOFFDATA

A SFCT tape for July may be created using TAPESA = '/CSM/ccm2/414/hist/h0104' and DAYSA = 030701.
I.4.5.6.2 OTSP

I.4.5.6. ->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 restricted to field pass 1, but any or all comparison Cases (*COMP) may share the same tape.

I.4.5.6.1. ->ITSP - Input of Time Series Plot Save Tapes

Primary ICPs: [SAVTSPR]

Secondary ICPs: <TSPPLcn> <TSPZLcn> <TSPALcn> <TSPZCcn> <TSPMCcn> <TSPFNPn> <TSPFPH> <TSLPASP> <TSLPSIZ> <TSZCASp> <TSZCSIZ> <TSMCAST> <TSMCSIZ> <CLTSLPc> <CLCTRGE> <CLCTRLT> <CLLABEL> <CLHIGH> <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> <TSPZCcn> <TSPMCcn>; contour intervals, scale factors, and dividing values for contour plots <TSPZCcn> <TSPMCcn>; aspect ratios, sizes, and colors for all plots <TSLPASP> <TSLPSIZ> <TSZCASp> <TSZCSIZ> <TSMCAST> <TSMCSIZ> <CLTSLPc> <CLCTRGE> <CLCTRLT> <CLLABEL> <CLHIGH> <CLLOWS>.

When a Time Series Plot Save Tape is input, the ICPs specifying the time series data <TSPPLcn> <TSPZLcn> <TSPALcn> <TSPZCcn> <TSPMCcn> should be assigned the same values (in the same order) as those in the run which wrote the save tape, except for the plot characteristics mentioned above.

I.4.5.6.2. ->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 input records; they will be needed again in the run that reads the
Save tape (*ITSP).

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).

I.4.5.7.1. ->OLSD - Output of LSD Save Tapes

Primary ICPs: [DRVRTYP] [SAVLS1c] [RTLS1c] [PKLS1c]

Secondary ICPs: <NSBTAPc> <NSBDAYc> <PWLS1c> <TITLS1c>

LSD processing (*LSDP) must be requested in order to output LSD Save Tapes [DRVRTYP]. Also required is a list of output LSD Save Tapes [SAVLS1c], a record ordering code [RTLS1c], and a packing density [PKLS1c]. A descriptive title <TITLS1c> and MSS write password <PWLS1c> are optional, as are the ICPs <NSBTAPc> and <NSBDAYc>, which provide alternate specification of the input tapes <TAPESc> and days <DAYSc> lists, respectively, in a generalized two-dimensional form appropriate for specifying ensembles.

A series of one or more LSD Save Tapes may be output for each comparison Case (*COMP). Each LSD Save Tape consists of a single master header record containing descriptive information, followed by an arbitrary number of data records. There are no embedded file marks. The first 50 words of each data record contain additional descriptive information, and the remainder of each data record contains the field values for a single horizontal slice. The descriptive information is documented in the tables at the end of this section.

There is a separate data record for each unique combination of the following three parameters: field level, Model time, and ensemble member. The order of the records is controlled in two independent ways. The first is by specifying...
the order in which these three parameters vary with respect to one another.
For example, if <RTLS1c> is set to 2, then field levels vary first (fastest),
time next, and ensemble member last (slowest); in other words, all field levels
are written for the first time, then all field levels for the second time, etc.
The order in which these three parameters vary within their respective domains
can also be controlled: the ordering of the ensemble members is determined by
the order in which the input tapes are specified with <NSBTAPc> or <TAPESc>;
the ordering of the Model times is determined by the order in which the days
are requested with <NSBDAYc> or <DAYSc>; and the order of the field levels is
determined (to some extent at least) by the order in which they are requested
with <FIELDcn>. For a particular three-dimensional field, the levels are
always ordered bottom to top; <FIELDcn> determines the order for fields read
from the input tape(s), but derived fields (*DFLD) follow input fields in the
order in which they are computed. Detailed ordering information is printed
when the LSD Save Tapes are written.

The total number of data records output (all Save Tapes) is determined by
the number of ensemble members, Model times, and fields being processed (i.e.,
all processed data is output). The number of data records on a single LSD Save
Tape is determined by the number of Save Tapes in the series (i.e., the records
are divided as evenly as possible between all tapes specified with [SAVLS1c]).
For all but the last tape in the series, the number of data records on each
Save Tape is equal to the total number of data records divided by the number of
tapes (using integer, or truncated division); that number of data records, plus
the remainder, if any, is written to the final tape in the series.

The horizontal slices of field values are written as Cray floating point
words in the following format:

* Longitude varies first, starting at the dateline (180 deg. E) and progressing
to the east. There are no longitudinal wraparound points.
* Latitude varies next from south to north. Latitudes are located on the Gauss-
ian grid (which is not linearly spaced).

These horizontal slices may be read into a two-dimensional array having
longitude as the first dimension and latitude as the second. The field values
(but never descriptive information) may be packed as requested by <PKLS1c>.
Packing is done with the NCAR SCD utility routine PACKA, which is also used by
both the Model and Processor for history tapes. The packing/unpacking routines
are not portable to other machines, and the data should not be packed if there
are any blocked points (*DEFS) (the packing algorithm would destroy the data).

The Processor has the capability for writing spectral data to List Sorted
Data (LSD) Save Tapes. This provides a simple means for making the Spherical
Harmonic Coefficients (SHC) available for further processing by users' own
programs. Whenever LSD Save Tapes are output <SAVLS1c>, the SHC are written
instead of gridpoint values, if spectral processing is requested <SPCcn> and
the value for the record type/ordering flag <RTLS1c> is negative. If spectral
time averaging is also requested <TIMAVGc>, then either the square of the norm
or the time variance of the SHC may be written instead of the SHC <SPCTAVV>.

The SHC are computed and normalized in the same manner as in the CCM; for
details of this computation, see pp. 27-31 of NCAR/TN-382+STR, "Description of
When spectral data are output to LSD Save Tapes, the format of the tapes is modified in two ways: 1) Words 16 through 19 of the miniheader (the first 50 words of each data record) are given new meanings; 2) the remainder of the data records contain one of several types of spectral values.

Words 16 through 19 of the miniheader are given the following new meanings.

16. \(M\) = Fourier wavenumber truncation for the data
17. \(N\) = largest associated Legendre Polynomial (ALP) for the data
18. \(K\) = largest ALP for any Fourier wavenumber
19. Flag indicating the format of the spectral data. This is one of the following two values, depending on the spectral time averaging options as described above: 102: The spectral data are linear SHC; each spectral value consists of two words: the real part immediately followed by the imaginary part. 112: The spectral data have no imaginary part and each spectral value consists of only one word. The data are either the square of the norm of the time average or the time variance, depending on the value of \(<\text{SPCTAVV}>\).

Note the important distinction between the terms spectral "value" and "word" in the above description of the format flag. If the data contain an imaginary part, then they may be read into a COMPLEX array dimensioned for the number of spectral values. Alternatively, the data may be read into a REAL array and the real and imaginary parts indexed separately. If, however, the data have no imaginary part, they should be read into an array typed REAL because there is only one word per spectral value.

The ordering of the spectral values within the data records is the same as in the internal arrays in the CCM, i.e., along diagonals. The following figure shows this ordering for a T6 spectral truncation (i.e., \(M=N=K=6\)). The numbers in the body of the figure indicate the ordering of the spectral values (where each value is either one or two words as described above).

LSD Save Tapes are written as direct access datasets (in COS pure binary format) in order to accomplish the reordering, but this does not prevent the Save Tapes from being subsequently read using sequential access utilities on the Cray. In other words, these Save Tapes may be read using standard Cray unformatted READ or BUFFER IN statements. The only discernible effect of the direct access writing is the fact that all records (including the header) are the same length.

The following two tables list the descriptive information in the master header records and data record mini-headers, respectively.
LSD Save Tape master header record

(all words are INTEGER unless otherwise noted)

<table>
<thead>
<tr>
<th>word number</th>
<th>description</th>
</tr>
</thead>
</table>

1. Record length in Cray words. All records on all tapes in the series have this length. It is the greater of the lengths of either the master header record \((12 + 2\times NLTOT + 10\times NTAPS)\) or the data record \((50 + NLON\times NLAT)\), where NLTOT is the total number of levels output (word 5), NTAPS is the number of Save Tapes in the series (word 10), NLON is the number of longitudes (word 14 of the data record), and NLAT is the number of latitudes (word 15 of the data record).

2. Data record order type code (as specified by <RTLS1c>). This type code indicates the order in which ensemble members, Model times, and field levels vary with respect to one another. The following table gives the order in which the parameters vary (fastest varying parameters are listed first) for each defined value of the type code.

<table>
<thead>
<tr>
<th>type code</th>
<th>parameter order</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>field level, time, ensemble member</td>
</tr>
<tr>
<td>4</td>
<td>time, ensemble member, field level</td>
</tr>
<tr>
<td>5</td>
<td>ensemble member, time, field level</td>
</tr>
</tbody>
</table>

3. Number of fields (not field levels) on the tape series
4. Number of free atmospheric levels for multilevel fields
5. NLTOT: total number of field levels on the tape series (all levels of all fields; e.g., a field defined at 18 levels in the vertical counts as 18 field levels)
6. Number of ensemble members on the tape series
7. Number of Model times on the tape series
8. Data packing density flag (1, 2, 3, or 4; a value of 1 indicates no packing)
9. Flag indicating the type of surface (vertical coordinate) on which the data are defined;
   1. sigma
   2. pressure
   3. height (deferred implementation)
   4. potential temperature
   5. sigma-pressure hybrid
10. NTAPS: number of Save Tapes in the series
11. Tape number (counter) for this Save Tape within the series
The remainder of the header record consists of the following three lists:

<table>
<thead>
<tr>
<th>first word</th>
<th>length</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>NLTOT</td>
<td>List of field names, one name for each field level (8 characters, left-justified, blank filled)</td>
</tr>
<tr>
<td>12+NLTOT</td>
<td>NLTOT</td>
<td>List of vertical level values, one value for each field level. The value is in units of millibars, degrees Kelvin, or is dimensionless (sigma/hybrid), depending on the vertical coordinate; it is multiplied by 1000 and converted to INTEGER.</td>
</tr>
<tr>
<td>12+2*NLTOT</td>
<td>10*NTAPS</td>
<td>List of full MSS pathnames, one name for each LSD Save Tape in the series. Each name consists of 80 characters (10 words), left-justified, null filled.</td>
</tr>
</tbody>
</table>

LSD Save Tape data record mini-header

(all words are INTEGER unless otherwise noted)

<table>
<thead>
<tr>
<th>word number</th>
<th>description</th>
</tr>
</thead>
</table>

1. Data record order type code (as specified by <RTLS1c>). This is the same as word 2 in the master header record.
2. Flag indicating whether or not the data vary within each of the following five data domains: ensemble members, Model times, field levels, latitudes, and longitudes. In other words, this flag indicates whether the Save Tape series contains only one ensemble member, or more than one; one Model time, or a series of time samples; etc. The flag consists of five independent decimal digits, with each digit corresponding to one domain. If a digit is 0, there is no variation within the corresponding domain; if a digit is 1, there is variation within the domain. The leftmost digit (ten thousands) always corresponds to the longitude domain and the next leftmost digit (thousands) always corresponds to the latitude domain. The correspondence for the remaining three digits (hundreds, tens, and units) varies according to the data record order type code: the rightmost (units) digit corresponds to the fastest varying domain, the next (tens) to the next fastest varying domain, and the third (hundreds) to the slowest varying domain. For example, the digit correspondence for record order type code 2 is:
For the field level domain, the flag is specific to the current field, i.e., it indicates if the current field is a single or a multilevel field. For the other four domains, the flag is universal, i.e., it indicates whether or not there is any variation within that domain with respect to the entire Save Tape series. Continuing the above example for record ordering type 2, suppose a) the output data has only one ensemble member at two Model times, b) the current field is a single-level field, and c) there are 40 latitudes and 48 longitudes; the flag value is then 11010 (decimal integer).

### Data Quality Flags

- **Flag digit domain**
  - ten thousands longitude
  - thousands latitude
  - hundreds field levels
  - tens Model time
  - units ensemble member

3. **Data quality flag (always set to 0 in the Processor)**

4. **Data packing density flag; same as word 8 in the master header record**

5. **Vertical coordinate flag; same as word 9 in the master header record**

6. **Vertical level value for the current field; same as in the second list in the master header record**

7. **Name of the current field (8 characters); same as in the first list in the master header record**

8. **Year of data (not used; unconditionally set to -1 in the Processor)**

9. **Month of data (not used; unconditionally set to -1 in the Processor)**

10. **Day of data (not used; unconditionally set to -1 in the Processor)**

11. **Current Processor day value <DAYS< truncated to an integer**

12. **Fractional part of current Processor day in seconds**

13. **Number of longitudes in the data record**

14. **Number of latitudes in the data record**

15. **Westernmost (first) longitude * 1000 (in degrees)**

16. **Easternmost (last) longitude * 1000 (in degrees)**

17. **Southernmost (first) latitude * 1000 (in degrees)**

18. **Northernmost (last) latitude * 1000 (in degrees)**

19. **First word of the horizontal slice containing field values (REAL)**

I-52
Ordering of SHC data along n-m diagonals for a T6 truncation

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>28</td>
<td>27</td>
<td>25</td>
<td>22</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>26</td>
<td>24</td>
<td>21</td>
<td>17</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>20</td>
<td>16</td>
<td>11</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>15</td>
<td>10</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>9</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

m - Fourier Wave Number
1.5 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 spectral derived fields (*SDFL).

1.5.1. GDFL - General description of derived fields

Primary ICPs: [FIELDcn]

Secondary ICPs: <TYPEc> <SPCcn> <DIFFLDn> <RATFLDn> <FLDSRCc> <SPCGRAD> <SPCVP> <DEFLDcn> <UNDEFDF>

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 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.

111 - during input from a CCM2, CCM1 or CCMOB hybrid coordinate history tape
11. during input from a CCMOB or CCM1 sigma coordinate history tape
12. during input from a CCMOA history tape
13 - during input from a pressure level history tape
132 - during input from a potential temperature level history tape
14 - during input from a Time Average or Time Series Save tape
20 - after input and before spectral operations (if requested) and before surface type masking.
21 - during grid point to spectral transformation
Computation types 111 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 six types, for a given Case. For this reason, a derived field can be defined for more than one computation type, but only if all the types are in the range 111 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 indicate that the field is computed in spectral space, i.e., it is a spectral derived field (*SDFL). It is necessary to request spectral transformation (*SPEC) 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 that request these derived fields.

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). (The Case merging option (*CMRC), however, provides a way around this restriction.) 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 Case A or B 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 (i.e., no longer processed). Derived fields may be based on time average statistics (*TAVG), as long as they are computed after time averaging is completed (*ORDR). Once computed, requested derived fields are processed along with all other fields. It is important to keep in mind that general transformation options process ALL fields being carried along at the time the transformation is performed, including fields which have not been explicitly requested. In most cases this is exactly what is desired, but when spectral processing is requested (*SPEC), this may be an undesirable side effect. See <SPCEFcn> for a solution to this problem.

Case C derived fields are requested <FIELDcn> and treated in the same way as those for Cases A and B with one important exception. If a Case C derived field is to be based on a field that is not another Case C derived field, then the "source" field must be explicitly requested in a manner which ensures that
it is placed in Case C when Case C is created. For example, if a Case C derived field is to be based on a Case difference field, it is necessary to explicitly request that difference field with the <DIFFLDn> ICP. If Case C is created with the Case merge option (*CMRG), then the required field must be explicitly requested for Case A or B. Since explicitly requested fields are not automatically deleted in such situations, it may be desirable to explicitly delete them as soon as they are no longer needed. This can be done with the ICP <DEFLDcn>, but it is the responsibility of the user to ensure that needed fields are not deleted too soon.

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 an arithmetic expression assigned to the ICP <DERFLD>. Most spectral derived fields (*SDFL) are, in a sense, a mixture: the computational algorithm is fixed by the code, but the field names are specified by the user. There are two classes of spectral derived fields: vector pair derived fields (*VPDF) <SPCVP>, and gradient derived fields (*GDFL) <SPCGRAD>. 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 share the same limitations on data availability: they must be based on longitude-height slices (i.e., different latitudes are not available simultaneously). Derived fields computed in spectral space have a different restriction: they must be based on longitude-latitude slices. Different Model times are not available for any derived field computations, but note that 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 and user-defined derived fields have an additional characteristic associated with them. The definitions for many of these derived fields contain algorithms (such as vertical finite differencing) that are valid only when the data are located on a particular type of surface (e.g., sigma, pressure, etc.). In order to prevent inappropriate application of these algorithms, each derived field is assigned a vertical coordinate flag that indicates on which type of surface(s) the field may be computed. Each of the 5 lowest order bits of this flag is assigned a particular surface type, and the flag is interpreted as follows (bit 1 is least significant):

- bit 1 on - OK to compute on sigma surfaces
- bit 2 on - OK to compute on pressure surfaces
- bit 3 on - ignored (reserved for constant height surfaces)
- bit 4 on - OK to compute on potential temperature surfaces
- bit 5 on - OK to compute on hybrid coordinate 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, and a value of 31 (decimal) means the field can be computed on any surface. Higher order bits are ignored.

The icp <UNDEFDF> specifies which code-defined derived fields (*CDFL) to be

I-56
undefined. Undefining fields with this ICP has the effect of freeing up the code defined name so that it can be used for other purposes, for example, defining a user-defined derived field (*UDFL). The default for <UNDEFDF> is that no fields are undefined.

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

I.5.2. ->CDFL - Code-defined derived fields

Primary ICPs: [FIELDcn]

Secondary ICPs: <TYPEc> <SPCcn> <DIFFLDn> <RATFLDn> <ABMERGE> <DEFLDcn> <UNDEFDF>

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). Code-defined derived fields may be "undefined" with <UNDEFDF>, allowing the field name to be used later. All code-defined derived fields are alphabetically listed below. Complete definitions follow.

<table>
<thead>
<tr>
<th>name</th>
<th>comp.</th>
<th>surf.</th>
<th>type(s)</th>
<th>flag</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHI</td>
<td>211</td>
<td>31</td>
<td></td>
<td></td>
<td>velocity potential</td>
</tr>
<tr>
<td>CONDH</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>condensational heating rate</td>
</tr>
<tr>
<td>DP</td>
<td>11</td>
<td>2</td>
<td></td>
<td></td>
<td>Hybrid coordinate pressure layer thickness</td>
</tr>
<tr>
<td>DELPRES</td>
<td>41</td>
<td>2</td>
<td></td>
<td></td>
<td>pressure layer thickness</td>
</tr>
<tr>
<td>DIV</td>
<td>21</td>
<td>31</td>
<td></td>
<td></td>
<td>horizontal wind divergence</td>
</tr>
<tr>
<td>DKE</td>
<td>61</td>
<td>31</td>
<td></td>
<td></td>
<td>specific kinetic energy of differences</td>
</tr>
<tr>
<td>DLNPSX</td>
<td>24</td>
<td>31</td>
<td></td>
<td></td>
<td>x-derivative of natural log of surface pressure</td>
</tr>
<tr>
<td>DLNPSY</td>
<td>24</td>
<td>31</td>
<td></td>
<td></td>
<td>y-derivative of natural log of surface pressure</td>
</tr>
<tr>
<td>DVMAG</td>
<td>61</td>
<td>31</td>
<td></td>
<td></td>
<td>velocity magnitude of differences</td>
</tr>
<tr>
<td>EDOTDPDE</td>
<td>31</td>
<td>16</td>
<td></td>
<td></td>
<td>CCM2 vertical velocity in hybrid coordinates</td>
</tr>
<tr>
<td>ENET</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td>net total atmospheric energy flux</td>
</tr>
<tr>
<td>ENETS</td>
<td>11,12</td>
<td>1</td>
<td></td>
<td></td>
<td>net surface energy flux</td>
</tr>
<tr>
<td>ENETT</td>
<td>11,12</td>
<td>1</td>
<td></td>
<td></td>
<td>net top of Model energy flux</td>
</tr>
<tr>
<td>HTO</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td>geopotential height, full levels, CCM08 version</td>
</tr>
<tr>
<td>HT1</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td>geopotential height, full levels, CCM1 version</td>
</tr>
<tr>
<td>KE</td>
<td>11-14</td>
<td>31</td>
<td></td>
<td></td>
<td>specific kinetic energy</td>
</tr>
<tr>
<td>KTOOPV0</td>
<td>31</td>
<td>1</td>
<td></td>
<td></td>
<td>kappa<em>temperature</em>omega/pressure, CCM08 version</td>
</tr>
<tr>
<td>KTOOPV1</td>
<td>31</td>
<td>1</td>
<td></td>
<td></td>
<td>kappa<em>temperature</em>omega/pressure, CCM1 version</td>
</tr>
<tr>
<td>LNPS</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td>natural log of surface pressure</td>
</tr>
<tr>
<td>MQ</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td>water vapor mass (3-dimensional)</td>
</tr>
<tr>
<td>NRADS</td>
<td>11,12</td>
<td>1</td>
<td></td>
<td></td>
<td>net radiative surface flux</td>
</tr>
<tr>
<td>PRECT</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td>total precipitation (accumulated)</td>
</tr>
<tr>
<td>PRES</td>
<td>11,12</td>
<td>1</td>
<td></td>
<td></td>
<td>pressure on sigma surfaces</td>
</tr>
<tr>
<td>PRESP</td>
<td>13,41</td>
<td>1</td>
<td></td>
<td></td>
<td>pressure on pressure surfaces</td>
</tr>
<tr>
<td>PSI</td>
<td>211</td>
<td>31</td>
<td></td>
<td></td>
<td>stream function</td>
</tr>
<tr>
<td>PSL</td>
<td>11,12</td>
<td>1</td>
<td></td>
<td></td>
<td>sea level pressure</td>
</tr>
<tr>
<td>QSRC</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td>moisture source term</td>
</tr>
</tbody>
</table>
RADD  11  1    net radiative heating rate
RELHUM 11-13 31 relative humidity
SIGDOTF 31 1  CCM1 sigma vertical velocity at full levels
SIGDOTH 31 1  CCM1 sigma vertical velocity at half levels
TCLD  11,12 1 total cloud fraction
TEST 11-13 31 programmable test field (used for debugging)
THETA 11,12 1 potential temperature on sigma surfaces
TMODK 11 1 temperature modified for computing KTOOPV1
TMQ 11 1 water vapor mass in a column (2-dimensional)
TV 11 1 virtual temperature
UD 24 31 u-velocity computed from divergence alone
UVSQ 23 31 total specific kinetic energy spectra
UVSQD 23 31 specific kinetic energy spectra (divergent part)
UVSQR 23 31 specific kinetic energy spectra (rotational part)
UZ 24 31 u-velocity computed from vorticity alone
VADVGV1 31 1 vertical advection of VADGIN, CCM1 version (see Note 1)
VADVQVO 31 1 vertical advection of moisture, CCM0B version
VADVQVI 31 1 vertical advection of moisture, CCM1 version
VADVTVO 31 1 vertical advection of temperature, CCM0B version
VADVTVI 31 1 vertical advection of temperature, CCM1 version
VADVUV1 31 1 vertical advection of U velocity, CCM1 version
VADVVV1 31 1 vertical advection of V velocity, CCM1 version
VARWV 61 31 specific kinetic energy variance
VD 24 31 v-velocity computed from divergence alone
VIKE 11 1 vertical integral of kinetic energy
VIME 11 1 vertical integral of potential energy
VITE 11 1 vertical integral of total energy
VMAG 11-14 31 horizontal wind speed
VOR 21 31 horizontal wind vorticity
VZ 24 31 v-velocity computed from vorticity alone
Z 11,12 1 geopotential height, half-levels (see Note 2)
Z1 11 1 geopotential height using CCM1 formulation
Z2 111 1 geopotential height using CCM2 formulation

Notes:

1) VADGIN is a "generic" input field; it must be created by the user prior to
    the computation of VADVGV1.

2) Z should be requested only when working with CCM0A, or if the values are to
    be compared with previously computed values of Z. In general, HT0 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 CCM0B. (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 interpo-
    lated 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.

Definitions of Code-Defined Derived Fields

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

\[ \text{CHI} = -\frac{(A^2 \times \text{DIV})}{(N \times (N+1))} \]

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

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

\[ \text{CONDH}(k) = \frac{G \times L \times QC}{C_p \times (\text{sig}(k+1)-\text{sig}(k))} \]

where \( G \) is the acceleration due to gravity (9.80616 m/s**2),
\( L \) is the latent heat of condensation (2.5104E6 J/kg),
\( C_p \) is the specific heat of dry air at constant pressure (1.00464E3 J/kg/K),
\( \text{QC} \) is the condensation rate (kg/m**2/s),
\( \text{PS} \) is the surface pressure (Pascals), and \( \text{sig}(k) \) and \( \text{sig}(k+1) \) are the values of sigma at the upper and lower layer interfaces, respectively, for the layer at which \( \text{CONDH}(k) \) is defined.

NOTE: In CCMOB, QC does not reflect the vertical redistribution of moisture associated with moist convective adjustment, so \( \text{CONDH} \) is not the local effective heating rate, although the vertical integral is correct. In CCM1, however, \( \text{CONDH} \) does represent the local heating rate.
>DELPRES
Description: pressure layer thickness
Computation Type(s): 41
Vertical Surface Type(s): pressure
Fields Required: PS
Units: Pa
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)

\[ DP(k) = 0.5 \times ( P(k-1)-P(k+1) ) \quad \text{for } P(k-1/2) < PS \]
\[ DP(k) = 0.0 \quad \text{for } P(k+1/2) > PS \]
\[ DP(k) = PS - P(k+1/2) \quad \text{for } P(k+1/2) < PS < P(k-1/2) \]

for k=NLVLS, (NLVLS>1)

\[ DP(NLVLS) = P(NLVLS-1/2) - PTOP \quad \text{(for specified PTOP)} \]
\[ DP(NLVLS) = \text{AMIN1} \left( P(NLVLS-1/2), P(NLVLS-1)-P(NLVLS) \right) \]
\[ \text{(for PTOP undefined)} \]

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,
\( P(k+1/2) = 0.5 \times ( P(k) + P(k+1) ) \),
\( P(1/2) > PS \) for all PS, and
PTOP may be specified via the ICP <PTOPc>

>DIV
Description: horizontal wind divergence
Computation Type(s): 21 (spectral processing required)
Vertical Surface Type(s): all
Fields Required: U, V
Units: 1/s
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see
"Description of the NCAR Community Climate Model (CCM2)",

>DKE
Description: specific kinetic energy of wind differences
Computation Type(s): 61
Vertical Surface Type(s): all
Fields Required: DU, DV (both must be requested as differences <DIFFLDn>)
Units: J/kg
Vertical Location: layer midpoints (full levels)
Algorithm: \[ \text{DKE} = 0.5 \times ( DU**2 + DV**2 ) \]
>DLNPSX
Description: x-derivative (partial) of natural log of surface pressure
Computation Type(s): 24 (spectral processing required)
Vertical Surface Type(s): all
Fields Required: LNPS (derived)
Units: Pa/m
Vertical Location: single level
Algorithm: computed in spectral space;

\[ DLNPSX = \frac{1}{A \cos(\phi)} \times \frac{D(LNPS)}{D(\lambda)} \]

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

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

\[ DLNPSY = \frac{1}{A} \times \frac{D(LNPS)}{D(\mu)} \]

where A is the radius of the earth
MU is the sine of the latitude
and \( \frac{D(\ )}{D(\ )} \) indicates a partial derivative

>DP
Description: Hybrid coordinate pressure layer thickness
Computation Type(s): 111
Vertical Surface Type(s): hybrid
Fields Required: PS
Units: Pascals
Vertical Location: layer midpoints (full levels)
Algorithm: DP = P(k-1/2) - P(k+1/2) (lower interface pressure minus the upper)

>DVMAG
Description: velocity magnitude of wind differences
Computation Type(s): 61
Vertical Surface Type(s): all
Fields Required: DU, DV (both must be requested as differences <DIFFLDn>)
Units: m/s
Vertical Location: layer midpoints (full levels)
Algorithm: DVMAG = sqrt( DU**2 + DV**2 )
>EDOTDPDE
Description: hybrid vertical velocity at half levels (layer interfaces)
Computation Type(s): 31
Vertical Surface Type(s): hybrid
Fields Required: U, V, DIV, DLNPSX, DLNPSY
   (DIV, DLNPSX, DLNPSY are spectral derived fields)
Units: 1/s
Vertical Location: layer midpoints (half levels)
Algorithm: See Eq. 3.a.15, "Description of the NCAR Community Climate Model (CCM2)", NCAR/TN-382+STR, June 1993.

>ENET
Description: net (total) energy flux into the atmosphere
Computation Type(s): 11
Vertical Surface Type(s): sigma
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
Vertical Surface Type(s): sigma
Fields Required: FRSA, FRLA, HFL, QFL (CCM0B)
   FRSA, FRLA, LHFLX, SHFLX, TSTAR (CCM0A)
Units: W/m**2
Vertical Location: single level
Algorithm (CCM0B):

   ENETS = FRSA - FRLA - ( HFL + RH20*HLAT*QFL ) / MAX(1,DT)

where RH20 is the density of water (1.0E3 kg/m**3)
   HLAT is latent heat of condensation (2.5104E6 J/kg)
   DT is accumulation time (sec) for HFL and QFL

Algorithm (CCM0A):

   ENETS = FRSA - FRLA - LHFLX - SHFLX

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

   ENETS = FRSA - FRLA - LHFLX - SHFLX - 1.046 * ( TSTAR + 1.06 )

>ENETT
Description: net energy flux into the top of the atmosphere
Computation Type(s): 11,12
Vertical Surface Type(s): sigma
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
Vertical Surface Type(s): sigma
Fields Required: T, PHIS
Units: m
Vertical Location: layer midpoints (full levels)
Algorithm: The geopotential height at level k is given by:

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

where k is the level index increasing downward from 1 at the highest layer to K at the lowest,
PHIS is the surface geopotential (Pa),
T(j) is the Model temperature at level j (K),
GO is 9.8 m/s**2,
R is the gas constant for air (287.04 J/kg/K ), and
B(k,j) is the matrix used by CCMOB for integrating the hydrostatic equation. See "Description of NCAR Community Climate Model (CCMOB)", NCAR/TN-210+STR, May 1983.

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

$$HT1(k) = \frac{PHIS}{GO} + \frac{R}{GO} \times \sum_{j=1,K} (B(k,j) \times TV(j))$$

where k is the level index increasing downward from 1 at the highest layer to K at the lowest,
PHIS is the surface geopotential (Pa),
TV(j) is the virtual temperature at level j (K),
GO is 9.8 m/s**2,
R is the gas constant for air (287.04 J/kg/K ), and
B(k,j) is the matrix used by CCM1 for integrating the hydrostatic equation. See "Description of NCAR Community Climate Model (CCM1)", NCAR/TN-285+STR, June 1987.

>KE
Description: specific kinetic energy
Computation Type(s): 111,11,12,13,132,14
Vertical Surface Type(s): sigma, hybrid
Fields Required: U, V
Units: J/kg
Vertical Location: layer midpoints (full levels)
Algorithm: $KE = 0.5 \times (U^{**2} + V^{**2})$
>KTOOPVO
Description: kappa*temperature*omega/pressure, CCMOB version
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: T, U, V, DIV, DLNPSX, DLNPSY
(DIV, DLNPSX, and DLNPSY are spectral derived fields)
Units: K/s
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
Vertical Surface Type(s): sigma
Fields Required: TMODK, U, V, DIV, DLNPSX, DLNPSY
DIV, DLNPSX, and DLNPSY are spectral derived fields;
TMODK (type 11 derived field) should be excluded from spectral processing <SPCEFcn> in order to avoid smoothing it.
Units: K/s
Vertical Location: full levels (layer midpoints)
Algorithm: KTOOPV1 at level k is given by:

KTOOPV1(k) = Kappa * TMODK(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 CCM1; it is given by:

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

I-64
where $\Delta\sigma(i)$ is delta sigma at full level $i$, and $B(j,k)$ is the CCM1 hydrostatic integration matrix; see Eqs. 3.b.20, 3.b.21, p. 41 of "Description of NCAR Community Climate Model (CCM1)", NCAR/TN-285+STR, June 1987.

>`LNPS`
Description: natural log of surface pressure
Computation Type(s): 111,11
Vertical Surface Type(s): sigma, hybrid
Fields Required: PS
Units: Pa
Vertical Location: single level
Algorithm: $LNPS = \ln(PS)$

>`MQ`
Description: water vapor mass (3-dimensional)
Computation Type(s): 111,11
Vertical Surface Type(s): sigma, hybrid
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
Vertical Surface Type(s): sigma
Fields Required: FRSA, FRLA
Units: W/m$^2$
Vertical Location: single level
Algorithm: $NRADS = FRSA - FRLA$

>`PRECT`
Description: total precipitation (averaged or accumulated)
Computation Type(s): 11,111
Vertical Surface Type(s): sigma, hybrid
Fields Required: PRECL, PRECC
Units: CCM1: m/s (average)
CCM08: m (accumulated)
Vertical Location: single level
Algorithm: $PRECT = PRECL + PRECC$

>`PRES`
Description: pressure on sigma or hybrid surfaces
Computation Type(s): 11,12,111
Vertical Surface Type(s): sigma or hybrid
Fields Required: PS
Units: Pa
>PRES
Description: pressure on pressure surfaces
Computation Type(s): 13, 41
Vertical Surface Type(s): pressure
Fields Required:
Units: Pa
Vertical Location: full pressure levels
Algorithm: PRES(K) = 100. * PLEV(K) (PLEV is on the header)

>PSI
Description: stream function
Computation Type(s): 211 (spectral processing required)
Vertical Surface Type(s): all
Fields Required: VOR (derived)
Units: m**2/s
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space;
\[ PSI = - \left( \frac{(A^2) \times VOR}{N \times (N+1)} \right) \]
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): 111, 11, 12
Vertical Surface Type(s): sigma, hybrid
Fields Required: PHIS, PS, T
Units: Pa
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:
\[ TSURF = T(NLEV) + SF \times ( T(NLEV-1) - T(NLEV) ) \]
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 \( S(\text{NLEV}) \) 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:

\[
\text{G/(R*GMOIST)}
\]

\[
\text{PSL=} \text{PS} \times (1 + \text{GMOIST} \times \text{PHIS}/(\text{TSURF} \times \text{G}))
\]

where \( \text{PS} \) is the surface pressure (Pa), \( \text{PHIS} \) is the surface geopotential \( (m^2/s^2) \), \( R \) is the gas constant for air \( (287.04 \text{ J/kg/K}) \), \( G \) is the acceleration due to gravity \( (9.80616 \text{ m/s}^2) \), and \( \text{GMOIST} \) is the moist adiabatic lapse rate \( (6.5 \times 10^{-3} \text{ K/m}) \).

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

\[
\text{QSRC} = (\text{QFL} - (\text{PRECL} + \text{PRECC})) \times \text{RHO}
\]

where \( \text{QFL} \) is evaporation (accumulated) \( (m) \), \( \text{PRECL} \) is the large scale precipitation (accumulated) \( (m) \), \( \text{PRECC} \) is the convective precipitation (accumulated) \( (m) \), and \( \text{RHO} \) is the density of water \( (1.0 \times 10^3 \text{ kg/m}^3) \).

>RADD
Description: net radiative heating rate
Computation Type(s): 111,11
Vertical Surface Type(s): sigma,hybrid
Fields Required: QRS, QRL
Units: K/day
Vertical Location: layer midpoints (full levels)
Algorithm: \( \text{RADD} = 8.64 \times 10^4 \times (\text{QRS} + \text{QRL}) \)

>RELHUM
Description: relative humidity
Computation Type(s): 111,11,12,13
Vertical Surface Type(s): sigma, pressure, hybrid
Fields Required: PS, T, MIXRAT (CCMOA), Q (CCM1 or CCMOB)
Units: per cent; \( T \) must be in K units, \( \text{PS} \) in Pa units
Vertical Location: layer midpoints (full levels)
Algorithm: Relative humidity is computed from the following equation:

\[
\text{RELHUM} = 100. \times \text{MIXRAT} \times \frac{\text{PS} \times \text{SIGMA} - \text{ES}}{0.622 \times \text{ES}}
\]
where MIXRAT is the mixing ratio (kg/kg),
PS is the surface pressure (Pa),
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 (Pa).

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
t line being drawn around areas of negative RELHUM, and avoids
bunching of negative contours around meaningless negative values.

>SIGDOTF
Description: sigma vertical velocity at full levels
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: U, V, DIV, DLNPSX, DLNPSY
(DIV, DLNPSX, DLNPSY are spectral derived fields)
Units: 1/s
Vertical Location: layer midpoints (full levels)
Algorithm: See Eqs. 3.29 - 3.32, pp. 23-24 of "Description of NCAR

>SIGDOTH
Description: sigma vertical velocity at half levels
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: U, V, DIV, DLNPSX, DLNPSY
(DIV, DLNPSX, DLNPSY are spectral derived fields)
Units: 1/s
Vertical Location: layer midpoints (half levels)
Algorithm: See Eqs. 3.b.6, 3.b.7, p. 39 of "Description of NCAR

>TCLD
Description: total cloud fraction (all layers combined)
Computation Type(s): 111,11,12
Vertical Surface Type(s): hybrid,sigma
Fields Required: CLD for CCM0A, CLOUD for CCM0B, CCM1, CCM2
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): 111,11,12,13,132
Vertical Surface Type(s): all
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 verifying 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): 111,11,12
Vertical Surface Type(s): sigma, hybrid
Fields Required: T, PRES (derived)
Units: K
Vertical Location: full sigma levels (layer midpoints)
Algorithm:

\[
\text{THETA}(k) = T(k) \times \left(1.5 / \text{PRES}(k)\right)^{R/CP}
\]

where \( R = 287.04 \, \text{J/kg/K} \)
and \( CP = 1.00464 \times 10^3 \, \text{J/kg/K} \)

>TMODK
Description: temperature modified for use in computing KTOOPV1
Computation Type(s): 111,11
Vertical Surface Type(s): sigma, hybrid
Fields Required: T, Q
Units: K
Vertical Location: full levels
Algorithm:

\[
\text{TMODK} = T \times \frac{1. + (Rv/R - 1.)Q}{1. + (Cpv/Cp - 1.)Q}
\]

where \( Rv = 461. \, \text{J/kg/K} \)
\( R = 287.04 \, \text{J/kg/K} \)
\( Cpv = 1.81 \times 10^3 \, \text{J/kg/K} \)
\( Cp = 1.00464 \times 10^3 \, \text{J/kg/K} \)

>TMQ
Description: water vapor mass in a column (2-dimensional)
Computation Type(s): 111,11
Vertical Surface Type(s): sigma, hybrid
Fields Required: MQ (derived)
Units: kg/m^2
Vertical Location: single level
Algorithm: TMQ = sum of all levels of MQ
>TV
Description: virtual temperature
Computation Type(s): 111,11
Vertical Surface Type(s): sigma, hybrid
Fields Required: T, Q
Units: K
Vertical Location: full levels
Algorithm:

\[ TV = T \times (1 + (\frac{Rv}{R} - 1)Q) \]

where \( Rv = 461 \) J/kg/K
\( R = 287.04 \) J/kg/K

>UD
Description: U-velocity computed from divergence alone
Computation Type(s): 24 (spectral processing required)
Vertical Surface Type(s): all
Fields Required: VOR, DIV (derived)
Units: m/s
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see "Description of the NCAR Community Climate Model (CCM2)", NCAR/TN-382+STR, June 1993.

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

\[ UVSQ = \frac{1}{4} \times \frac{N(N+1)}{(A^2)} \times (\overline{\text{PSI}} \times \text{PSI} + \overline{\text{CHI}} \times \text{CHI}) \]

where \( \text{PSI} = -\left(\frac{(A^2) \times \text{VOR}}{N \times (N+1)}\right) \)
\( \text{CHI} = -\left(\frac{(A^2) \times \text{DIV}}{N \times (N+1)}\right) \)
\( A \) is the radius of the earth
\( N \) is the degree of the Legendre polynomial
and the overbar indicates the complex conjugate.

>UVSQD
Description: Specific kinetic energy spectra (divergent part)
Computation Type(s): 23 (spectral processing required)
Vertical Surface Type(s): all
Fields Required: VOR, DIV (derived)
Units: J/kg
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space;
UVSQD = \frac{1}{4} \times \frac{N(N+1)}{(A^2)} \times (\text{CHI} \times \text{CHI})

where

\text{CHI} = - \frac{(A^2) \times \text{DIV}}{(N \times (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: Specific kinetic energy spectra (rotational part)
Computation Type(s): 23 (spectral processing required)
Vertical Surface Type(s): all
Fields Required: VOR, DIV (derived)
Units: J/kg
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space;

UVSQR = \frac{1}{4} \times \frac{N(N+1)}{(A^2)} \times (\text{PSI} \times \text{PSI})

where

\text{PSI} = - \frac{(A^2) \times \text{VOR}}{(N \times (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)
Vertical Surface Type(s): all
Fields Required: VOR, DIV (derived)
Units: m/s
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see
"Description of the NCAR Community Climate Model (CCM2)",

> VADVGV1
Description: vertical advection of generic field VADGIN, CCM1 version.
VADGIN may be any full-level field created by the user.
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: VADGIN, SIGDOTH (spectral derived field)
Units: units of VADGIN per second
Vertical Location: layer midpoints (full levels)
Algorithm: See Eqs. 3.b.1 - 3.b.6, p. 39 of "Description of NCAR

> VADVQVO
Description: vertical advection of moisture, CCMOB version
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: Q, SIGDOTH (spectral derived field)
I.5.2 CDFL

Units: kg/kg/s
Vertical Location: layer midpoints (full levels)
Algorithm: See Eqs. 3.15 - 3.27, pp. 22-23 of "Description of NCAR Community Climate Model (CCM02)", NCAR/TN-210+STR, May 1983.

> VADVQ1
Description: vertical advection of moisture, CCM1 version
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: Q, SIGDOTH (spectral derived field)
Units: kg/kg/s
Vertical Location: layer midpoints (full levels)

> VADTV1
Description: vertical advection of temperature, CCM1 version
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: T, SIGDOTH (spectral derived field)
Units: K/s
Vertical Location: layer midpoints (full levels)

> VADVU1
Description: vertical advection of eastward velocity, CCM1 version
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: U, SIGDOTH (spectral derived field)
Units: m/s**2
Vertical Location: layer midpoints (full levels)

> VADVW1
Description: vertical advection of northward velocity, CCM1 version
Computation Type(s): 31
Vertical Surface Type(s): sigma
Fields Required: V, SIGDOTH (spectral derived field)
Units: m/s**2
Vertical Location: layer midpoints (full levels)
>VARWV
Description: variance of wind vector
Computation Type(s): 61
Vertical Surface Type(s): all
Fields Required: U, V, KE (KE derived, all must be time averaged)
Units: J/kg
Vertical Location: same as U, V
Algorithm: \( \text{VARWV} = 2.0 \times \text{KE} - \bar{U}\bar{U} - \bar{V}\bar{V} \)

>VD
Description: V-velocity computed from divergence alone
Computation Type(s): 24 (spectral processing required)
Vertical Surface Type(s): all
Fields Required: VOR, DIV (derived)
Units: m/s
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see "Description of the NCAR Community Climate Model (CCM2)", NCAR/TN-382+STR, June 1993.

>VIKE
Description: vertical integral of kinetic energy
Computation Type(s): 11
Vertical Surface Type(s): sigma
Fields Required: U, V, PS
Units: J/m²
Vertical Location: single level
Algorithm: \( \text{VIKE} = \text{sum of all levels of KE}(K) \)

where \( \text{KE}(K) = 0.5 \times (\bar{U}\bar{U} + \bar{V}\bar{V}) \times D\Sigma(K) / G \)

and \( D\Sigma(K) = \text{SIGMA}(K+1/2) - \text{SIGMA}(K-1/2) \)
K is vertical index, increasing downward
G is gravitational acceleration (9.80616 m/s²)

>VIPE
Description: vertical integral of potential energy
Computation Type(s): 11
Vertical Surface Type(s): sigma
Fields Required: T, PHIS, PS
Units: J/m²
Vertical Location: single level
Algorithm: \( \text{VIPE} = \text{sum of all levels of PE}(K) \)

where \( \text{PE}(K) = (\text{CP} \times T + \text{PHIS}) \times D\Sigma(K) / G \)
and \( D\Sigma(K) = \text{SIGMA}(K+1/2) - \text{SIGMA}(K-1/2) \)
K is vertical index, increasing downward
CP is specific heat of dry air at constant pressure
(1.00464E3 J/kg/K)
G is gravitational acceleration (9.80616 m/s²)
>VITE
Description: vertical integral of total energy
Computation Type(s): 11
Vertical Surface Type(s): sigma
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): 111, 11, 12, 13, 132, 14
Vertical Surface Type(s): all
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
Vertical Surface Type(s): all
Fields Required: U, V
Units: 1/s
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see
"Description of the NCAR Community Climate Model (CCM2)",

>VZ
Description: V-velocity computed from vorticity alone
Computation Type(s): 24
Vertical Surface Type(s): all
Fields Required: VOR, DIV (derived)
Units: m/s
Vertical Location: layer midpoints (full levels)
Algorithm: computed in spectral space; see
"Description of the NCAR Community Climate Model (CCM2)",

>Z
Description: Geopotential height (also see HTO and HT1)
Computation Type(s): 11, 12
Vertical Surface Type(s): sigma
Fields Required: PHIS, T, Q (CCM08 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 im-
material for the purpose of vertical interpolation because of the ex-
trapolation 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:

\[ Z(\text{NLEV}) = \frac{\text{PHIS}}{G}, \]

where PHIS is the surface geopotential (Pa), 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:

\[ Z(k) = Z(k+1) + R \cdot B(k) \cdot TV(k)/G, \]

where \( R \) is the gas constant for air (287.04 J/kg/K ),
\( B(k) = \ln\left( \frac{\text{SIGMA}(k)}{\text{SIGMA}(k+1)} \right), \)
\( \text{SIGMA}(k) \) is the Model sigma value at level \( k \), and
\( TV(k) \) is the virtual temperature given by:

\[
TV(k) = \frac{0.622 + Q}{0.622 \cdot (1 + Q)} \quad \text{(for CCM08)}
\]

\[
TV(k) = T(k) \quad \text{(for CCM0A)}
\]

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.

> Z2
Description: Geopotential height using the CCM2 Hydrostatic formulation.
Computation Type(s): 111
Vertical Surface Type(s): hybrid
Fields Required: PHIS, T, Q
Units: m

Vertical Location: layer interfaces (half-levels).
Algorithm: The CCM2 surface geopotential height is computed using the
hydrostatic equation given by equation 3.a.109 in "Description
of the NCAR CCM2", NCAR TN/382+STR.
1.5.3. UDFA - User-defined derived fields

Primary ICPs: [DERFLD] [FIELDcn]

Secondary ICPs: <DEFLDcn> <UNDFDF>

It is possible to define derived fields by specifying the computational algorithm as an arithmetic expression on an ICP data input record [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 (a field or a scalar constant) or an operator (a 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 stack. This is similar to the method used by many hand calculators, except that, here, operands may be 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. 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.

The fields on which user-defined derived fields are based may be other derived fields of any computation type, as long as the basis fields are computed first. User-defined derived fields of the same computation type are computed in the order in which they are defined, not the order in which they are requested. All code-defined derived fields (*CDFL) are computed prior to the computation of user-defined derived fields of the same computation type.

The icp <UNDEFDF> specifies which code-defined derived fields (*CDFL) to be undefined. Undefining fields with this ICP has the effect of freeing up the code defined name so that it can be used for defining a user-defined derived field. The default for <UNDEFDF> is that no fields are undefined.

Although user-defined derived fields cannot be computed in spectral space, the gradient (*GRDF) and vector-pair derived field capabilities (*VPDF) allow some degree of control in defining new derived fields to be computed in spectral space (*SPEC).
I.5.4 SDFL
November 1993
I.5.4.1 VPDF

I.5.4. Speculative derived fields

The computation of derived fields in spectral space (*SPEC) is handled differently than computations in gridpoint space. Although there is no generalized procedure for user-specification of a computation algorithm, there are a number of fixed algorithms which can be applied to arbitrary fields. There are two separate classes of algorithms: computation of horizontal gradients (*GDF), and computations involving vector component pairs of fields (*VPDF). These two classes of spectral derived fields are described separately in the following two topics.

I.5.4.1. Vector-pair derived fields

Primary ICPs: [SPCVP] [FIELDcn] [SPCcn]
Secondary ICPs: <DERFLD>

There are a number of derived fields based on the velocity component fields U and V, which may be computed in spectral space (*SPEC). Since the names as well as the computational algorithms for these fields are fixed by the code, they are actually code-defined derived fields (*CDFL). These fields, however, are also a special case of a more general class of derived fields called "vector-pair" derived fields. It is possible to define an arbitrary pair of fields as a vector pair, and apply the same (U,V) pair algorithms to create new derived fields with names defined by the user. Specifically, the algorithms available for computing vector-pair derived fields are those used to compute DIV, VOR, CHI, PSI, UD, UZ, VD, and VZ from U and V. See (*CDFL) for a description of these fields and algorithms.

The vector-pair derived fields are defined by the ICP [SPCVP], which specifies one or more additional vector pairs (i.e., in addition to the U,V pair), 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 <DERFLD>. 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 be computed only if [SPCcn] is set to 'YES'. Note, however, that defining fields as a vector pair changes the way they are handled in spectral space: they receive the same special treatment given to the U,V pair, and are therefore subject to the same restrictions. See (*SPEC) for a discussion of this special treatment.
1.5.4.2. GRDF - Gradient derived fields

Primary ICPs: [SPCGRAD] [FIELDcn] [SPCcn]

Secondary ICPs: <DERFLD>

The code-defined derived fields DLNPSX and DLNPSY are the gradient components of the log of surface pressure (LNPS) computed in spectral space. The same gradient computation algorithm (see (*CDFL)) can also be applied to an arbitrary field to generate its gradient components, which are referred to as "gradient derived fields". The ICP [SPCGRAD] is used to name the gradient components, and also to specify the field from which they are computed. The ICP [SPCVP] just defines vector-pair derived fields; these fields are computed only if they are requested with the [FIELDcn] ICP. Since the gradients are computed in spectral space, it is also necessary to request spectral processing (*SPEC) [SPCcn]. The field for which the gradient is to be computed may be either an input field or a previously-computed derived field < DERFLD >.

I.6. VERT - Vertical interpolation

Two forms of vertical interpolation are available: from sigma or hybrid to pressure surfaces (*VPRS), and from sigma or hybrid to potential temperature surfaces (*VPOT). Only one of these interpolation options can be requested in a given jobstep. The controlling ICPs for the two options are very similar, but they are completely independent. The algorithms are also very similar; in fact, most of the code is common to both. The next section fully describes pressure interpolation, then the following section describes the differences for isentropic interpolation. Note that all forms of vertical interpolation in the Processor treat all fields as scalars; in other words, there are no vector transformations for such quantities as "horizontal" wind velocity.

There is a subtle problem involving the surface pressure field PS (which is required in order to locate the sigma/hybrid levels) when the horizontal resolution is changed in spectral space. See (*SINT) for a discussion.

I.6.1. 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 vertical coordinate to an arbitrary set of pressure levels [PRESSLE] (same levels for all fields). When interpolating to pressure surfaces, surface pressure must be explicitly 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 pressure
3 - linear in natural log of natural log of pressure

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.

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 <NLCDP> 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 that 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.

I.6.2 VPOT - Vertical interpolation to potential temperature surfaces

Primary ICPs: [TEMPLEV] [FIELDcn]

Secondary ICPs: <INTDT> <LBTDT> <NLCDT> <TINTXL> <TINTMLT> <MXLNSCL> <LXLNSCL>

Fields on Model surfaces can be vertically interpolated to an arbitrary set of potential temperature surfaces. This interpolation uses the same technique, and has basically the same (independent) options as the interpolation to pressure surfaces (*VPRS): the ICP <TEMPLEV> is used to request this option and specify the isentropic surfaces, while <INTDT>, <LBTDT>, <NLCDT>, and <TINTXL> are used to specify the default interpolation type, lower boundary treatment, number of levels to copy, and the field exceptions list, respectively. Interpolation to isentropic surfaces, however, has an additional, unique option: <TINTMLT> to control the handling of multiple isentropic surfaces (i.e., the case where a requested potential temperature surface occurs at more than one elevation within a given vertical column).

Isentropic interpolation has the same surface pressure availability requirements as pressure interpolation. In addition, the derived field THETA (potential temperature) (*CDFL) must be explicitly requested for each Case being processed (<FIELDAn> and/or <FIELDBn>). THETA is defined for computation types 11 and 12 only, which means that it can only be computed from history tape input. This implies that if input is from a Processor Save Tape, isentropic vertical interpolation can only be done if THETA is on the Save Tape. Note, however, that a Save Tape may be translated into a history tape in a preliminary jobstep (*OHST); THETA can then be computed from the translated history tape as long as both temperature (T) and surface pressure (PS) are available.

The field THETA is not interpolated onto potential temperature surfaces, and is unconditionally deleted immediately after other fields are interpolated. If subsequent processing of potential temperature on potential temperature surfaces is necessary (e.g., in order to generate a field containing potential temperature for use in a derived field expression), THETA may be copied to a user-defined derived field prior to interpolation (*UDFL).

Fields interpolated to potential temperature surfaces may be output to any type of Processor Save Tape (*SAVT), including history tapes. When a history tape containing data on potential temperature surfaces is input to the Processor, the ICP <TYPEc> should be assigned the value 'THETA'.

For fields interpolated to potential temperature surfaces, the associated level word contains the potential temperature in degrees Kelvin, followed by a 'T' suffix. When requesting vertical cross-section plots (*PMEX) (*PLAX), it
may be desirable to override the default logarithmic scaling for the vertical coordinate \(<MXLNSCL> <LXLNSCL>\).

Only fields on sigma surfaces can be vertically interpolated; requesting both isentropic and pressure interpolation results in a fatal error.

As with pressure interpolation, isentropic interpolation is linear with respect to a transformed sigma vertical coordinate, with the form of the transform determined by the interpolation type code. This method requires knowledge of the vertical variation of \(THETA\) in order to determine a transformed sigma value on the requested potential temperature surfaces. It is assumed that \(THETA\) varies linearly with respect to the natural log of sigma (i.e., within a given sigma layer), regardless of the interpolation type code specified for any field. If a requested potential temperature surface is outside of the range of \(THETA\) in a particular column, then sigma on this surface is computed by extrapolating \(THETA\) above the highest (or below the lowest) level available, but only if the highest (or lowest) available layer is thermodynamically stable. Otherwise all fields are blocked (*DEFS) for that particular surface. All fields are also blocked on any theta surface whose equivalent sigma value is greater than 5 (this may occur for very low potential temperatures, especially in the tropics).

I.7 ->SPEC - Spectral processing

Primary ICPs: [SPCcn]

Secondary ICPs: <SPCINTc> <SPCBPcn> <SPCDFcn> <SPCEFcn> <SPCMNKc> <TIMAVGc> <SPCTAVV>

The data being processed may be transformed into Spherical Harmonic Coefficients (SHC) [SPCcn], and in this form, may be spatially interpolated in the horizontal (*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 that 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 other than roundoff error. For many fields, however, the two transforms will have the effect of filtering the data. Note that this effect is not limited to requested fields; fields being processed solely for the purpose of computing derived fields (*DFLD) later may also be affected. 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 are transformed into SHC. The transformation algorithm is defined by the pentagonal spectral truncation parameters (M, N, and K) currently associated with the gridpoint data. Alternate truncation parameters for this transformation may be specified with <SPCMNKc>; this
ICP, however, does not affect the subsequent transformation back into gridpoint space, as does <SPCINTc>.

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. At the default CCM2 T42 resolution, M, N and K all equal 42. For additional details, see pp. 41-45 of NCAR/TN-382+STR, "Description of NCAR Community Climate Model (CCM2)", June 1993.

Since the horizontal velocities U and V are not properly defined by the spectral transformation, they are automatically replaced in spectral space by vorticity (VOR) and divergence (DIV). This special handling of U, V, VOR, and DIV generally results in the appropriate processing of these fields, but CAUTION is advised because it is sometimes necessary to consider the subtleties involved in a particular processing request. Since computing VOR and DIV requires both U and V, the spectral transformation cannot be properly done for U or V alone. If only one or the other is requested, an error termination results. If VOR and/or DIV are being processed in gridpoint space at the time spectral transformation occurs, they are treated as scalars (i.e., handled individually). If they are spectrally smooth (which is normally the case) they are not affected by the transformation. If U and V are also being processed, an ambiguity arises because there are two sources for VOR and/or DIV in spectral space. This situation should be avoided; it is not well handled by the code, and the results are to some extent unpredictable. A similar problem can arise if the other (U,V) vector-pair derived field (*VPDF) names (CHI, PSI, UD, UZ, VD, and VZ) are used for some alternate purpose. If additional vector pairs are defined with <SPCVP>, all of the above restrictions and potential problems also apply to that pair and the associated derived fields.

<SPCTAVV> specifies the final form of data produced by spectral time averages; either the time average or time variance of the Spherical Harmonic Coefficients or the linear Spherical Harmonic Coefficients themselves.

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

I.7.1. ->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. This option differs from that provided by <SPCMNK> in that the subsequent gridpoint resolution may also change, and consistent values for the horizontal grid resolution must be specified for [SPCINT].

There is a subtle problem associated with changing the horizontal resolution of the data if vertical interpolation is to be subsequently performed. The surface pressure field PS is required in order to locate the sigma surfaces for the interpolation. If PS is not explicitly requested <FIELDcn>, it is automatically read from the input tape(s) and set aside for use later in the vertical interpolation. It is NOT automatically spectrally interpolated to the appropriate horizontal resolution along with the rest of the fields, so a fatal error results unless the problem is resolved explicitly by the user. The simplest solution is to explicitly request PS so that it is spectrally interpolated to the appropriate resolution; this procedure, however, has the effect of smoothing PS prior to the spectral interpolation, because it is the natural log of PS which is spectrally smooth in the Model, not PS itself. Another solution is to spectrally interpolate the log of PS, then convert it back to PS for the vertical interpolation; this requires more than one Processor jobstep. Yet another solution is to perform the vertical interpolation first, then do the spectral interpolation; this also requires more than one Processor jobstep. It is probably best to resolve this problem based on careful consideration of the specific application; the Processor generates a fatal error if PS is not available at a consistent horizontal resolution.

I.7.2 SBND - Spectral bandpass filtering

Primary ICPs: [SPCcn] [SPCBPcn] [FIELDcn]

Secondary ICPs: none

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

\[
\begin{align*}
m &= 0 \text{ for } m < M \text{ or } m > M \\
n &= 0 \text{ for } n < N \text{ or } n > N
\end{align*}
\]

where m is the Fourier wavenumber, n is the order of the Legendre polynomial, M, M', N, and N are input parameters [SPCBPcn]. 1 2 1 2

All SHC are set to zero outside of a RECTANGLE defined by these input parameters. This bandpass rectangle need not be wholly contained within the spectral truncation polygon; the only requirement is that SOME values of (m,n) must be common or a fatal error results. Since the CCM1 spectral representation is pentagonal, the specified (m,n) limits must be chosen with care. For
example, if filtering is desired only in the zonal direction, then it is necessary to set the maximum passed value of \( n \) at least as large as \( K \) (since in general, \( K > N \)). See Figure 1, p. 29, NCAR/TN-382+STR, "Description of the NCAR Community Climate Model (CCM2)"; June 1993.

I.8 \(-\) TAVG - Time average statistics

Primary ICPs: \([\text{TIMAVGc}] \quad [\text{TAPESc}] \quad [\text{DAYSc}] \quad [\text{FIELDcn}]\]  

Secondary ICPs: \(<\text{SDFLDcn}> \quad <\text{CVFLDcn}> \quad <\text{TCFLDcn}> \quad <\text{PRFLDcn}> \quad <\text{ZSTFLcn}> \quad <\text{ZCFLcn}> \quad <\text{DERFLD}> \quad <\text{DEFLDcn}>\]

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

When Time Average Save tapes are input (*STAV), the input tapes \([\text{TAPESc}]\) define the time sample to be averaged, and the days list \([\text{DAYSc}]\) is ignored. A time series consisting of only one time sample may be averaged. 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. See Appendix B for some examples.

The time average statistics that 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.

\[ \overline{x} \quad \text{time average } [\text{TIMAVGc}] \]

\[ \sqrt{x'x'} \quad \text{time standard deviation } <\text{SDFLDcn}> \]

\[ x'y' \quad \text{time covariance } <\text{CVFLDcn}> \]

\[ \overline{xy} \quad \text{time average of product } <\text{PRFLDcn}> \]
x*y* : time average of product of deviations from zonal mean
\langle TCFLDcn \rangle

sqrt[x*x*] : time average of zonal standard deviation \langle ZSTFLcn \rangle

[x*y*] : time average of zonal covariance \langle ZCVFLcn \rangle

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>. It is also possible to correlate fields from different Cases by first combining the Cases (*CMRG).

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.

If a gridpoint is blocked (*DEFS) for ANY time sample included in the average, then the time average is blocked at that point. It is possible, however, to prevent this blocking and compute a meaningful time average by using a complex technique described in the "Event Statistics" example in Appendix B.

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

I.9 ->COMP - Case comparison

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

Secondary ICPs: <RATFLDn> <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. (See (*CMRG) for additional computation possibilities). The fields to be compared should be defined at the same levels in the vertical, with the possible exception of extra bottom levels (*XBTL); these levels 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; 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 (see the "Time and Space Correlation Statistics" examples in Appendix B).

When Case differences are requested [DIFFLDn], a printout of "significant" differences may also be requested <NSDPRT>. 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 some processing of Cases A and B occurs before Case C is created (*ORDR), there are a few option limitations.
I.10 CMRG  

1.10 ->CMRG - Case merging

Primary ICPs: [ABMERGE] [TAPEsc] [DAYSc] [FIELDcn] [TYPEc]

Secondary ICPs: <TITLEc> <DEFLDcn>

Two different sets of data (Cases A and B) can be requested for processing in the same run, and all Case A fields may be merged with all Case B fields to form Case C [ABMERGE]. This merging occurs at the point where Case C would be created if differences or ratios (*COMP) were requested (*ORDR). If merging is requested, no differences or ratios can be computed, Cases A and B must be compatible with respect to resolution in time and space, and no Case B field may have the same name as any Case A field. (Fields may be effectively renamed by a user-defined derived field definition (*UDFL).) Case A descriptive information is used for Case C. Merged Cases need not have the same number of levels as long as Case B consists solely of single level fields.

This option provides a more general alternative to the simple Case Comparison option (*COMP). When combined with user-defined derived fields (*UDFL), a wide variety of comparison statistics can be computed. See the "Time and Space Correlation Statistics" and "Significance Statistics" examples in Appendix B.

I.11 MASK  

1.11 ->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 1.0E36 (*DEFS); these points are then excluded 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 that may perform a similar masking as part of another operation. Also see (*HORA).

The distribution of surface types must be available whenever masking is requested. The same surface type distribution is used for all Cases (*COMP), and it is assumed to be independent of time. (But see (*UDFL) and the [DERFLD] example for a more general method of masking.) There are three possible sources for surface type information:

1) a Surface Type Save Tape (*SFCT) <SFCTTAP>,
2) the first input history tape for Case A,
3) the first input history tape for Case B.

This list of possibilities is examined in the listed order and the first located source is used, with one exception: if a Surface Type Save Tape is to be created <SFCTCRT>, the source cannot be another Surface Type Save Tape (<SFCTTAP> is ignored if specified). In other words, a Surface Type Save tape can be created only from a history tape source. If a Surface Type Save Tape is input, it must have the same horizontal resolution as the Cases being processed, or a fatal error results.

Two T42 Surface Type Save Tapes, containing the sea ice distributions for January and July from the CCM2 414 control run, have been archived on the MSS, /CCMPROC2/ccm2/414/sfct/030101 and /CCMPROC2/ccm2/414/sfct/030701. The two Surface Type Save Tapes for R15 data are: /CCMPROC/SR15JA and /CCMPROC/SR15JL.

I.12 -> 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:

\[
FF = \sum_{j=1,n}^{i} W \cdot UF_{i-(n/2)+j-1}
\]

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) and (n/2) is truncated to an integer, and
- W is the jth filter weight.

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:
I.14 ZEST

November 1993

* 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
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 severe for filters approaching this limit. See (*MLIM) for a discussion
of memory limitations. These limitations apply to Cases A and B individually
since Case A is filtered completely before filtering of Case B begins.

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

1.13 \rightarrow 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, requested fields [FIELDcn]
should be limited to those involved in the zonal eddy statistic computations;
no other time average statistics can be computed. When requested, these statis-
tics [ZSTFLcn] [ZCVFLcn] are computed for all levels of the specified fields,
and are automatically plotted as meridional cross sections (*PMEX). See (*TAVG)
for definitions of the statistics that may be requested. The zonal eddy statis-
tics are 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 statis-
tics 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.

I.14 ->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);
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 spatial averaging options are Case independent, i.e., they apply to all Cases being processed.

There are, in general, two possible ways to proceed after computing spatial averages. After plotting, the averages may be discarded, and processing can continue using the full, three-dimensional fields. Alternately, the full, three-dimensional fields can be discarded, and processing can continue using the spatial averages as the starting point for subsequent operations. For zonal, meridional, and vertical averaging, which of these approaches is taken is determined by a secondary option specified independently for each type of average. The options available for subsequent processing of spatial averages are somewhat limited (e.g., spectral processing is not allowed, and horizontal projection plots cannot be drawn for zonal averages), but all averages can be the basis for user-defined derived fields (*UDFL), and the averages can be written to most types of Save Tapes (*SAVT). A particular type of spatial average cannot be computed more than once on the same data; if zonal or meridional averages are written to a Save Tape, vertical cross-section plots may be requested in subsequent runs with the ICPs <MXPLOT> and <LXPLOT>, respectively.

Each type of spatial average is discussed individually in the next four topics.

I.14.1 ->VERA - Vertical averaging

Primary ICPs: [VERAVG]

Secondary ICPs: <VAVGDSP> <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 either the vertical averages or the unaveraged data <VAVGDSP> 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 specified <VBKFR>.

When computed, vertical averages may be plotted as horizontal projections (*PHOR) <VAVGDSP> <HPROJ>. 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.

The vertical average is computed by using the trapezoid rule to integrate between the boundaries of the specified (or defaulted) range <VAVRNG>. This procedure has the effect of weighting the value at each level according to the layer thickness, but only half a layer is used at the endpoints. Since ground elevation values are never averaged, this implies that part of the free atmosphere immediately above the ground is always excluded from the average. That part of the atmosphere above the highest level available is also always excluded.

This averaging scheme can be easily modified by changing the integration weights computed in subroutine VAVWTS in Deck VERAVG. This subroutine has sufficient information for a fairly complex scheme. Another technique that does not require modifying the code is the use of the user-defined derived field (*UDFL) functions .DSWVSUM, .VSUM, or .DSTIMES <DERFLD>. Also see the "Vertical Integration" example in Appendix B for a more general technique.

Vertical averages cannot be computed more than once on the same data; if the averages are written to a Save Tape (*SAVT), horizontal projection plots may be requested in subsequent runs with the ICP <HPROJ>.

I.14.2. ->ZONA - Zonal averaging

Primary ICPs: [ZONAVG]

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

When zonal averaging is requested [ZONAVG], all levels of all fields are averaged along each latitude line with all longitudes weighted equally. Processing then continues, with subsequent operations using either the zonal averages or the unaveraged data <ZAVGDSP> as a starting point (*SPAV) (*ORDR). 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 may be plotted as meridional cross sections (*PMEX) <ZAVGDSP>. Plot characteristics can be controlled through the use of the appropriate ICPs (*PLOT).

Zonal averages cannot be computed more than once on the same data; if the averages are written to a Save Tape (*SAVT), vertical cross-section plots may be requested in subsequent runs with the ICP <MXPLOT>.

I.14.3. ->MERA - Meridional averaging

Primary ICPs: [MERA AVG]

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

When meridional averaging is requested [MERA AVG], 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. Processing then continues, with subsequent operations using either the meridional averages or the unaveraged data <MAVGDSP> as a starting point (*SPAV) (*ORDR). 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 may be plotted as latitudinal cross sections (*PLAX) <MAVGDSP>. Plot characteristics can be controlled through the use of the appropriate ICPs (*PLOT).

Meridional averages cannot be computed more than once on the same data; if the averages are written to a Save Tape (*SAVT), vertical cross-section plots may be requested in subsequent runs with the ICP <LXPLT>.

I.14.4. ->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.

I-92
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), and they cannot be otherwise output or form the basis for any other operations. This surface type masking 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. The same surface type distribution is used for all Cases (*COMP), and it is assumed to be independent of time. (But see (*UDFL) and the [DERFLD] example for a more general method of masking.) There are three possible sources for the surface type information: 1) a Surface Type Save Tape (*SFCT) <SFCTTAP>, 2) the first input history tape for Case A, and 3) the first input history tape for Case B. This list of possibilities is examined in the listed order and the first located source is used, with one exception: if a Surface Type Save Tape is to be created <SFCTCRT>, the source cannot be another Surface Type Save Tape (<SFCTTAP> is ignored if specified). In other words, a Surface Type Save Tape can be created only from a history tape source. If a Surface Type Save Tape is input, it must have the same horizontal resolution as the Cases being processed, or a fatal error results.

There are two Surface Type Save Tapes archived on the MSS for use with R15 resolution data. They are /CCMPROC/SR15JA and /CCMPROC/SR15JL, containing the sea ice distributions for January and July, respectively. The two T42 Surface Type Save Tapes for CCM2 are: /CCMPROC2/ccm2/414/sfct/030101 and 030701.

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.

See (*UDFL) and the examples for <DERFLD> for an alternate method of masking, using an arbitrary, time-dependent, input field. Also see the "Event Statistics" example in Appendix B.
Primary ICPs: none

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

Since the Processor is designed to be run in a batch mode, execution of the graphics code does not result in the direct generation of plots, but rather in the production of a file containing graphics instructions (metacode). The first graphics subtopic (*PDIS) discusses the disposition of this metafile, i.e., its transfer to another computer for archival and/or translation into a graphical image. The following subtopics describe the Processor's graphics capabilities by categorizing the plots 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 (*SPGR). 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. Finally, there is a general topic on specifying plot color (*PCLR). Although the individual plot type topics describe which parts of each type of plot can be set to different colors, this general discussion provides an overview of the color capabilities common to all plot types.

Some Processor options (such as spatial averaging) may 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 CCM). 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: hybrid values are multiplied by 1000 and have a suffix of "H"; sigma values are multiplied by 1000 and have a suffix of "S"; pressure levels are in millibars and have a suffix of "P"; potential temperature levels are in degrees Kelvin and have a suffix of "T"; and ground and surface levels are indicated by "1000.C" 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 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 file containing the plots <ICPECHO>.

I.15.1. ->PDIS - Disposition of plot file

Primary ICPs: [DPLTMF]

Secondary ICPs: <DPLTRCP> <DPLTCP> <DPLTCA> <DPLTCT> <DPLTIT>
<DPLTXT> <DPLTFN> <MNFRMS> <MXFRMS> <INDEX>
<ICPECHO> <NUMPLT> <MSPFXO>

Under most circumstances, disposition of the plot file is handled automatically by the Processor code. 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, the standard Processor attempts to send any plots that have been made to the fiche camera.

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 II for the details.

[DPLTMF] - one or more mainframes (Network nodes) to receive the plot file(s);
<DPLTRCP> - destination filename for rcp to a remote UNIX host;
<DPLTCP> - destination filename for cp to a local SCD host;
<DPLTCA> - camera type for Dicomed disposes;
<DPLTCT> - Case title to be used as Dicomed plot title if <DPLTIT> is not specified;
<DPLTIT> - plot title for Dicomed disposes;
<DPLTFN> - destination filename;
<DPLTXT> - TEXT field for DISPOSE;
<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 input records 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 are meaningful only to the Dicomed. This text is added using the NCAR printer simulation package PRSIM.

The plots produced by one or more jobsteps (*DEFS) may be broken up into two or more groups that are disposed separately. This is done automatically if the
size of the plot file approaches the maximum size that 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 DISPLOT procedure in the standard run deck (*CJDK).

The MSS is a valid destination for the plot file (DPLTMF = 'MS'), but it requires some special handling with respect to file naming. The full file pathname (MSPN) is formed as follows: use the Case-independent prefix specified (or defaulted) for <MSPFXO> to obtain the full directory pathname; then append the filename specified with <DPLTFN> (or the string "PLOTS" if <DPLTFN> is not specified) followed by two numeric digits giving the plot file dispose group. See <DPLTFN> for examples. Note that it is necessary to incorporate the plot dispose group number into the MSPN since the MSS would overwrite files disposed by previous jobsteps if the names were the same. (This is not a problem with other dispose destinations.) The MSS retention time and write password cannot be specified for execution-time disposes (system defaults are always used), and plot files are never saved as permanent datasets on the Cray disks. If necessary, these features can be implemented by disabling execution-time disposes (as described below) and calling "mswrite" from the batch script.

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 batch script 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 records are always copied to the dataset CMETAIE. 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 CMETAIE is copied (immediately after being written) to dataset CMETAP using the Dicomed printer simulator package PRSIM. It is also unconditionally copied to the print file. All of the plotting instructions (metacode) are written directly to the dataset CMETAP as they are generated. As each frame is completed, a plot description line is added to the index dataset CMETAI, 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 CMETAI to GMETAP, if requested by the ICP <INDEX>. Next, gmeta is copied to GMETAP, and CMETAP is then disposed to all requested mainframes, in the order in which they are specified. CMETAI, CMETAP, and GMETAE are then rewound, completing the dispose procedure. No end-of-files are ever written to any of these datasets.

I.15.2. ->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). Rectangular grid projections may be 'blown up', i.e., magnified to show a limited area (*LAHP). Polar regions in polar stereographic projections can be magnified by limiting the outer latitude boundary.

I.15.2.1. ->PHRE - Plotting rectangular horizontal projections

Primary ICPs: [HPROJ]

Secondary ICPs: <HPPTVAL> <HPMCINT> <HPSCAL> <HPCDIV> <HPCINT> <HPLFPVn> <HPVOPT> <HPVSCAL> <HPVDIN> <DASHLIN> <HPCNWKI> <HPDTCNT> <HPSMTH> <HPSMTHSL> <CLCNTNT> <CONHILO> <CONZERO> <HPRSIZE> <HPRASPR> <CONLABB> <CONLABS>

When rectangular horizontal projection plots are requested [HPROJ], each level of each field being processed is projected onto a horizontal surface (using a cylindrical equidistant projection) and plotted separately. Distances along the axes are proportional to latitude or longitude, and the latitude-longitude grid is rectangular, but areas are exaggerated at high latitudes. By default, the entire globe is projected onto a single frame, with both the left and right edges of the plot corresponding to 180 degrees East longitude, where grid point values are always defined and represented. The top and bottom edges of this default global 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. There are also options for producing plots that cover a limited area (i.e., less than the full globe); see the subtopic (*LAHP) for a discussion. Either geographic or Model (box) continental outlines may be drawn <HPCNWKI>, and their color can be controlled <CLCNTNT> and solid or dotted continental outlines with <HPDTCNT>. 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) <HPLFPVn>. Contours of vector magnitude may be superimposed over the vector arrows <HPVOPT>. <HPRSIZE> and <HPRASPR> can be used to specify the size and aspect ratio of the rectangular plot.

For contour plots (*PCON), the contour interval <HPCINT>, scale factor for
numeric labels <HPSCAL>, and a dividing value for contour line attributes
<HPCDIV> <DASHLIN> may all be controlled individually for each level of each
field. Specific contour intervals may be specified with <HPMCINT>. The plot-
tting of H's and L's (including values) at local maxima and minima is optional
<CONHILO>, as is the drawing of the zero contour level <CONZERO>. The integer
and real work arrays used by CONPAK can be modified with <HPCNWKR> and
<HPCNWKI>, respectively. Spline smoothing of contours can be requested and set
with the <HPSMTHSL> and <HPSMTH> ICPs.

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).

I.15.2.1.1.  -> LAHP - Limited area horizontal projections

Primary ICPs: [HPROJ] [HPRNDIV] [HPRBNDS] [HPRADP]

Secondary ICPs: <HPPTVAL> <HPCINT> <HPSCAL> <HPCDIV> <DASHLIN>
<HPLFVPN> <HPVOPT> <HPVSCAL> <HPVDIN> <HPCONOL>
<HPCNWKR> <HPCNWKI> <HPDTCNT> <HPSMTH> <HPSMTHSL>
<CONHILO> <CONZERO> <CLCNTNT>

There are two options for plotting cylindrical equidistant projections
(HPROJ = 'RECT') over limited (less than global) areas. In the first option,
which is the default, the ICP <HPRNDIV> is used to specify the number of times
the globe is to be divided (i.e., cut in half), forming individually plotted
sections. In other words, setting HPRNDIV = n results in the globe being
divided into 2**n plot frames. The first halving is always along a north-south
line, the second is along an east-west line, and so on, alternating between the
two. If the number of halvings is even, the sections are (approximately) twice
as wide as they are high; if n is odd, the sections are (approximately) square.
Since <HPRNDIV> defaults to 0, a global plot is produced if this ICP is not
explicitly set.

This global sectioning option is particularly useful for obtaining plots of
printed values <HPPTVAL>, because, at most common CCM resolutions, not every
gridpoint value can be printed if a single plot covers the entire globe. Since
point values are not printed along the plot borders, the boundaries of each
section are moved outward to the next gridpoint to provide some overlap between
sections (provided this does not result in the longitude span exceeding 360
degrees). This overlap is also added on contour plots so that both types of
plots have the same boundaries. Since the plot boundaries are always at grid-
point locations, individual sections do not necessarily contain the same number
of gridpoints (i.e., they may vary in their latitude and longitude spans). The
latitude and longitude scaling, however, is the same for all sections, so that
hardcopies of the sections may be cut and pasted together. If this is to be

I-98
done with contour plots, it is usually best to specify a contour interval <HPCINT> so that all sections have the same contour lines. If this option is used in conjunction with the printed value plot option <HPPTVAL>, the following values will result in the minimum number of plot sections without skipping any point values: for R15, use HPRNDIV=3; for T31 and T42, use HPRNDIV=5.

The second limited area plot option allows the specification of a number of arbitrary latitude-longitude rectangles, with each rectangle plotted on a separate frame. This type of plot is requested by specifying the plot boundaries with the ICP <HPRBNDS>, which overrides the global sectioning option (<HPRNDIV> is ignored). The dateline (180 degrees longitude) may be inside the specified rectangle(s). Specified plot boundaries are NOT moved to coincide with gridpoint locations. Contours extend to the outermost gridpoints within or on the borders; printed values extend to the outermost gridpoints, which are at least half a gridpoint away from the borders. Latitude and longitude are scaled identically, so the plot shape is determined by the specified limits. Each plot is made as large as possible, and is centered on the frame. When using this option, it is usually best to carefully choose the boundaries relative to gridpoint locations, especially if there is a high degree of magnification; otherwise there may be an unacceptably large margin of space between the edge of the contoured region and the plot borders.

Note that <HPRNDIV> and <HPRBNDS> qualify the <HPROJ> ICP, which must be set to 'RECT' in order for either of these ICPs to have any effect. All of the ICPs controlling the plot characteristics for global rectangular projections (*PHOR) also apply to limited area rectangular projections. Limited area plots can be produced for polar projections using the [HPRADP] (*PHPO).

I.15.2.2. ->PHPO - Plotting polar horizontal projections

Primary ICPs: [HPROJ]

Secondary ICPs: <HEMIS> <HPCINT> <HPCDIV> <HPSCAL> <DASHLIN>
<HPLFPVn> <HPVOPT> <HPSCAL> <HPVDIN> <HPCONOL>
<HPCNWKI> <HPCNKRI> <HPDTCNT> <HPSMTH> <HPSMTHSL>
<CONHILD> <CONZERO> <CLCNTNT> <HPRADP>

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. Either geographic or model (boxed) continental outlines may be drawn <HPCONOL>, 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) <HPLFPVn>. The latitudinal radius of the polar projection plots can be specified with <HPRADP>. 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 specific 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 contour 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>. The plotting of H's and L's (including values) at local maxima and minima is optional <CONHILO>, as is the drawing of the zero contour level <CONZERO>.

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).

[HPRADP] determines the cutoff latitude at the plot's perimeter; this is currently defaults to plot 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 that is linear with respect to latitude is used to determine field values at the plot perimeter.

I.15.3. PMEX - Plotting meridional cross sections

Primary ICPs: none

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

For multilevel fields, a meridional cross-section plot uses latitude as the abscissa and the height coordinate (sigma, pressure, or potential temperature) 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> if the vertical coordinate is sigma or pressure. 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 that 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 contour line attributes <MXCDIV> <DASHLIN> may all be set individually for each field. The plotting of H's and L's (including values) at local maxima and minima is optional <CONHILO>, as is the drawing of the zero contour level <CONZERO>. 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).

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). If the data have already been zonally averaged in a previous jobstep, then <MXPLOT> may be used to plot the averages. The range of latitudes to be plotted may also be specified <MXLTRNC>.

For multilevel fields, a latitudinal cross-section plot uses longitude as the abscissa and the height coordinate (sigma, pressure, or potential temperature) 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. <LXLNRNG> can be used to specify the longitude range of the plot. 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> if the vertical coordinate is sigma or pressure. 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 that 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 contour line attributes <LXCDIV> <DASHLIN> may all be set individually for each field. The plotting of H's and L's (including values) at local maxima and minima is optional <CONHILO>, as is the drawing of the zero
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). If the data have already been meridionally averaged in a previous jobstep, then <MXPLOT> may be used to plot the averages.

I.15.5. →PTIM - Plotting time series

There are currently five different types of time series that 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 ensemble processing has been requested <ENSMBLc>. In both of these kinds of runs, the time coordinate is labelled from one 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.

I.15.5.1. ->PTFL - Plotting time vs. field value

Primary ICPs: [TSPPLcn]

Secondary ICPs: <TSPFNPn> <TSPFPH> <TSLPASP> <TSLPSIZ> <CLTSLPc>
<DASHLIN> <TSPDYSc> <TSPYSCL>

This type of time series plot is a line plot (*PLIN), with time as the abs-
cissa 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 <TSPFNPn>, 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 has the
minimally expanded range that encompasses both individual ranges. The size
<TSLPSIZ> and aspect ratio <TSLPASP> used for all time series line plots may be
specified. <TSPYSCL> sets the y scaling for all time series line plot to
either linear or logarithmic. <TSDYSc> allows the days to be renumbered on all
time series plots.

I.15.5.2. ->PTZL - Plotting time vs. zonal average

Primary ICPs: [TSPZLcn]

Secondary ICPs: <TSPFNPn> <TSPFPH> <TSLPASP> <TSLPSIZ> <CLTSLPc>
<DASHLIN> <ZAVRNG> <ZBKFR> <TSPYSCL> <TSDYSc>

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 <TSPFNPn>, 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. <TSPYSCL> sets the y scaling for all time series line plot to either linear or logarithmic. <TSDYSc> allows the days to be renumbered on all time series plots.

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>.

I.15.5.3. -->PTAL - Plotting time vs. masked area average

Primary ICPs: [TSPALcn]

Secondary ICPs: <TSPFNPn> <TSPFPH> <TSLPASP> <TSLPSIZ> <CLTSLPc> <DASHLIN> <SFCTTAP> <MSKAVPR> <MSKAP> <MSKAPS> <MSKAZS> <TSPYSCL> <TSDYSc>

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, 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 <TSPFNPn> 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. <TSPYSCL> sets the y axis scaling for all time series line plots to either linear or logarithmic. <TSDYSc> allows the days to be renumbered on all time series plots.
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.

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.

I.15.5.4. ->PTZC - Plotting zonal average contours (time vs. latitude)

Primary ICPs: [TSPZCcn]

Secondary ICPs: <TSZCASP> <TSZCSIZ> <TSPZCLR> <DASHLIN> <ZAVRNG> <ZBKFR> <CONHILO> <CONZERO> <TSPYSCL> <TSDYSc>

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, the line label scale factor, and the contour dividing value [TSPZCcn]. <TSPYSCL> sets the y scaling for all time series line plot to either linear or logarithmic. <TSDYSc> allows the days to be renumbered on all time series plots. <TSPZCLR> specifies the latitude range to be plotted.

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). The plotting of H's and L's (including values) at local maxima and minima is optional <CONHILO>, as is the drawing of the zero contour level <CONZERO>.

I.15.5.5. ->PTMC - Plotting meridional avg. contours (long. vs. time)

Primary ICPs: [TSPMCcn]

Secondary ICPs: <TSMCASP> <TSMCSIZ> <TSPMCLR> <DASHLIN> <MAVRNG> <MBKFR> <CONHILO> <CONZERO> <TSDYSc>
This type of time series plot is a contour plot (*PCON) with longitude as the abscissa and time (increasing downward) 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 "Hovmöller 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]. <TSDYSsc> allows the days to be renumbered on all time series plots. <TSPMCLR> specifies the longitude range to be plotted.

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). The plotting of H's and L's (including values) at local maxima and minima is optional <CONHILO>, as is the drawing of the zero contour level <CONZERO>.

1.15.6. ->SPGR - Spectral graphics

Primary ICPs: [SPCcn] [SPSNCRF]

Secondary ICPs: <SPGYINT> <TIMAVGc>

If spectral processing is requested [SPCcn], then the square of the norm may be graphed for all spectral fields [SPSNCRF]. The ordinate on these graphs is the common log of the sum over Fourier wavenumber of the SHC, times its complex conjugate. The abscissa is the common 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.

1.15.7. ->PVAL - Plotting printed values

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

Secondary ICPs: <CLPNTVL> <HPRNDIV> <HPRBNDS>

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". If the point density on a particular plot is too high to print the values for every gridpoint, then points are skipped as appropriate. This problem can be eliminated on horizontal projection plots by requesting limited area plots (*LAHP) using <HPRNDIV> or <HPRBNDS>. Printed
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, continental outlines are superimposed. The color (*PCLR) of the printed value characters can be specified <CLPNTVL>.

I.15.8. ->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>.

I.15.9. ->PCON - Plotting contours

Primary ICPs: none

Secondary ICPs: <CONHILO> <CONZERO> <DASHLIN> <CLCTRGE> <CLCTREQ> <CLCTRLT> <CLHIGHS> <CLLOWS> <HPSMTH> <HPSMTHSL> <CONLABB> <CONLABS>

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. The plotting of H's and L's (including values) at local maxima and minima is optional <CONHILO>, as is the drawing of the zero contour level <CONZERO>. "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). A limited amount of control over the range of contour levels can be obtained by limiting fields with the user-defined derived field (*UDFL) functions .MIN and .MAX <DERFLD>, and plotting the result.

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.

The boxing of High and Low labels on all contour plots is controlled by <CONLABB> while the interval at which labels are drawn on the contour lines may be set with <CONLABS>.

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>. Spline smoothing of contours can be requested and set with the <HPSMTHSL> and <HPSMTH> ICPs.

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 that is linear with respect to distance on the frame (this is not necessarily linear with respect to the physical coordinate).

I.15.10. ->PVEC - Plotting vectors

Primary ICPs:  [HPLFPVn]

Secondary ICPs: <HPVOPT> <HPVSCL> <HPVDIN> <CLVLTD> <CLVGED> <HPDIV>

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 mag-
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 <HPDIV> can be specified <CLVGED> independently of the color for vectors whose magnitude is less than the dividing value <CLVLTD>.

I.15.11. ->PCLR - Plotting in color

Primary ICPs:  [COLOR]
Secondary ICPs:  <CLCNTNT> <CLCTRGE> <CLCTREQ> <CLCTRLT>
<CLHIGHS> <CLLLWOS> <CLPNTVL> <CLVLTD> <CLVGED>
<HPDIV> <HPCDIV> <MXCDIV> <LXCDIV> <CLTSLPc>
<TSPZCc> <TSPMCc> <DASHLIN> <CLBACK> <CLTABLE>

Since the plot utilities used by the Processor do not contain provisions for color filling of areas, 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 value of [COLOR] determines whether color plots will be produced by default. Often the lines on color plots may look clear on graphics monitors do not replicate well on color printers.

Colors are specified using an integer variable 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.

The metafile output by the Processor always contains instructions to change colors, with all colors for all plots set to the default color of 1. 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) <CLCNTNT>. Also, the color specified by <CLCTRGE> is used on both contour and some line plots.
Part II. Descriptions of Input Control Parameters (ICPs)

All ICP keywords are listed alphabetically below. Suffixes for case and field pass are represented by a lower case 'c' and 'n' respectively.

**ABMERGE**
Switch controlling the merging of fields from Cases A and B to form Case C (one CHARACTER value, 'YES' or 'NO', default is 'NO'). If 'YES' is specified, all requested fields for Case A are combined with all requested fields for Case B, and the result is processed as Case C. No differences and/or ratios (*COMP) may be requested in the same jobstep, Cases A and B must have the same spatial resolution, and no Case B requested field may have the same name as any Case A requested field at the time the merge takes place (Fields may be effectively renamed using user-defined derived fields (*UDFL)).

**BPHSTc**
Determines whether a History Save Tape (*OHST) can be written when there are blocked points in any of the output data fields (*DEFS) (one CHARACTER value, '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 Processor does not check for blocked points when computing derived fields of types 111, 11, 12 and 13. Also, a history tape containing blocked points cannot be input to the Model. If <PKHSTc> is larger than 1, then <BPHSTc> is ignored (packing a blocked point does not make sense because the packing algorithm will modify the blocked point value such that it will not be handled properly by the Processor on input).

**CLBACK**
Specify the default background and foreground colors for plots (one CHARACTER value, default is CLBACK = 'WONB').

CLBACK = 'WONB' - White lines on a Black background
CLBACK = 'BONW' - Black lines on a White background

Note: Plots produced with the 'BONW' options are not visible on microfiche.

**CLCNTNT**
Color (*PCLR) of continental outlines on horizontal projection plots (*PHOR) (one INTEGER value, default is CLCNTNT = 1).

**CLCREQ**
Color (*PCLR) of plotted contours (*PCON) equal to the dividing value <HPCDIV> <MXCDIV> <LXCDIV> <TSPZCcn> <TSPMCcn> (one INTEGER value, default is CLCREQ = 1). When set, this value overrides
CLCTRGE Color (*PCLR) of plotted contours (*PCON) greater than or equal to the dividing value <HPCDIV> <MXCDIV> <LXCDIV> <TSP2Ccnn> = CLCTRGE (one INTEGER value, default is CLCTRGE = 1). This is also the line color for line plots (*PLIN). See CLTABLE for a listing of the default color table values.

CLCTRLT Color (*PCLR) of plotted contours less than the dividing value <HPCDIV> <MXCDIV> <LXCDIV> <TSP2Ccnn> = CLCTRLT (one INTEGER value, default is CLCTRLT = 1). See CLTABLE for a listing of default color table values.

CLHIGHS Color (*PCLR) of all highs (H's and values) on all contour plots (*PCON) (one INTEGER value, default is CLHIGHS = 1). See CLTABLE for a listing of default color table values.

CLLABEL Color (*PCLR) of labels and borders on all plots (*PLOT) (one INTEGER value, default is CLLABEL = 1). See CLTABLE for a listing of default color table values.

CLLows Color (*PCLR) of lows (L's and values) on all contour plots (*PCON) (one INTEGER value, default is CLLows = 1). See CLTABLE for a listing of default color table values.

CLPNTVL Color (*PCLR) of point values on printed value plots (*PVAL) (one INTEGER value, default is CLPNTVL = 1). See CLTABLE for a listing of default color table values.

CLTABLE Sets color table values (four REAL values). The first value is the color table number and the following three are the Red, Green and Blue intensities. For example:

\[ \text{CLTABLE} = 0., 0.75, 0.15, 0.15 \]

results in the plots having a dark red background.

Colors 2-12 are reserved for the user to specify. By default they are set to white. Colors 13-23 are hardwired to the colors listed below and colors 24-256 are also available for the user.

Default Color Table Values:

<table>
<thead>
<tr>
<th>Number</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Black</td>
</tr>
<tr>
<td>1</td>
<td>White</td>
</tr>
<tr>
<td>2-12</td>
<td>White</td>
</tr>
<tr>
<td>13</td>
<td>Blue</td>
</tr>
<tr>
<td>14</td>
<td>Sky blue</td>
</tr>
<tr>
<td>15</td>
<td>Cyan</td>
</tr>
<tr>
<td>16</td>
<td>Green</td>
</tr>
<tr>
<td>17</td>
<td>Yellow</td>
</tr>
<tr>
<td>18</td>
<td>Red</td>
</tr>
<tr>
<td>19</td>
<td>Magenta</td>
</tr>
<tr>
<td>20</td>
<td>White</td>
</tr>
<tr>
<td>21</td>
<td>Brown</td>
</tr>
<tr>
<td>22</td>
<td>Orange</td>
</tr>
<tr>
<td>23</td>
<td>Black</td>
</tr>
<tr>
<td>24-256</td>
<td>User Definable</td>
</tr>
</tbody>
</table>

II-2
CLTSLPc Color (*PCLR) of all Case c lines for all time series line plots (*PTIM) (one INTEGER value, default is CLTSLPc = 1). See CLTABLE for a listing of the default color table values.

CLVGED Color (*PCLR) of plotted vectors (*PVEC) greater than or equal to the dividing value <HPVDIV> (one INTEGER value, default is 1). See CLTABLE for the default color table values.

CLVLTD Color (*PCLR) of plotted vectors (*PVEC) whose magnitude is less than the dividing value <HPVDIV> (one INTEGER value, default is CLVLTD = 1). See CLTABLE for a list of default color table values.

COLOR Controls automatic color plotting in the Processor (one INTEGER value, default is COLOR = 0 (black and white)). If non-zero, COLOR is the size of the color table used for auto-generation of a color spectrum across contour plots. The lowest contour line will be blue, then transitioning cyan, green, yellow, until red is used for the highest contour. For a 256 color workstation, COLOR=200 works well, which takes into account the 10-20 colors used by the workstation for displays and tools.

To build an N value color table, divide N values along the path: black-blue-cyan-green-yellow-red-magenta-white.

CONHILO Controls the marking of highs and lows (maxima and minima) on all contour plots (*PCON) (one CHARACTER value, 'YES' or 'NO', default is CONHILO = 'YES')

'YES' - local maxima and minima are marked with "H" and "L" symbols, respectively, on all contour plots. The field value associated with the high or low is also printed below the symbol. (default)

'NO' - highs and lows are not marked on any contour plots.
CONLABB  Controls the boxing of High and Low labels on all contour plots (*PCON).  See 'HLB' in the NCAR Graphics Guide to New Utilities (one INTEGER value, default is CONLABB = 0).

CONLABB = 0 - High and low labels are not boxed at all.
CONLABB = 1 - Perimeter of the box is drawn (in the same color as the label) after the label is drawn.
CONLABB = 2 - The box is filled before the line is drawn.
CONLABB = 3 - Both 1 and 2 above.

CONLABS  Controls the interval at which labels are drawn on the contour lines on all contour plots (*PCON) (one INTEGER value, default is 3).

For example:
CONLABS = 1 - every contour level is labeled.
CONLABS = 2 - every other contour level is labeled.

CONZERO  Controls the drawing of the zero contour level on all contour plots (One CHARACTER value, 'YES' or 'NO', default is CONZERO = 'YES').

'YES' - the zero contour level is drawn if zero is within the range of field values being contoured (default).
'NO' - the zero contour level is not drawn under any circumstances.

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

(covariance = 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 same Case [TIMAVGc]. The fields needed to compute the covariances must be explicitly requested via the [FIELDcn] ICP, however, 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 ICPs 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 CHARACTER value, 'YES' or 'NO', default is DASHLIN = 'YES').

'YES' - Contour lines less than the dividing value, and Case B time
series lines are both dashed.

'NO' - No lines are dashed on any plots.

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). The specified day values are interpreted as either absolute or relative to the start of Model integration or as dates <DAYTYPc>. The list may be specified in either of the following two forms:

1) 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 position on the input tapes, and they may be freely interspersed with actual day values. 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.

2) 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 REAL values, 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 ensemble processing (*ENSB) for details.
5. See <DAYTYPc> for examples of day specifications using dates.
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).

DAYTYPc

Specifies how the "day" values assigned to the ICPs <DAYSc> or <DLDAYSc> are to be interpreted (one CHARACTER value, default is DAYTYPc = 'RELATIVE'). With one exception (see below), this ICP is ignored if input is not from a history tape; for input from other types of Save Tapes (*SAVT), the specified "day" values must be consistent with the <DAYTYPc> specification for the Processor run which wrote the Save Tape.

Exception: If do-loop notation is used (<DAYSc> or <DLDAYSc>) with dates, then <DAYTYPc> should be set to 'DATE' regardless of the type of tape being input; otherwise the do-loop expansion may be incorrect.

'RELATIVE' - The day values are relative to the "base day" for the run, i.e., the "base day" is subtracted from the requested day before being compared to the "current day" on the history tape header. (default)

'ABSOLUTE' - The requested day is compared to the "current day" without first subtracting the "base day" on the history tape header.

'DATE' - The day values are interpreted as dates in the form YMMDD where YY, MM, and DD are a 2-digit year, month, and day, respectively. For example, 880315. is March 15, 1988. All years are relative to 1900, so YY becomes three digits after 1999; for example, January 1, 2000 is given by 1000101.. All values specified for <DAYSc> must be REAL (with fractions indicating partial days). The special, negative flag values for <DAYSc> may be intermixed with dates, and do-loop notation may also be used, with the starting and ending values both specified as dates, and the increment expressed as the number (possibly fractional) of 24-hour periods between desired dates. Note that <LEAPY> controls the handling of leap years. Specified dates are matched with the "current" dates on the history tape headers. All "day" values printed by the Processor (including on plots) are also in the date form. The "day" values written to all output Save Tapes (except History Save Tapes) (*SAVT) are also in date form, so subsequent input of
these tapes requires a consistent date form for the
<DAYS> values. Note: When using a "do-loop" date list
for <DAYS> or <DLDAYS>, both the starting and ending
dates specified should always be valid dates; otherwise
the expansion of the do-loop may be incorrect.

'DATE' - Same as for 'DATE' above, except that the header "cur-
rent date" is ignored and the requested dates are
matched against a date computed from the "base date"
and the "current day" (relative Model day). CAUTION:
Use of this option may result in relabelling of Model
time samples in a manner inconsistent with the solar
position used in the Model's radiation parameteriza-
tion.

Example 1:
DAYTYPA = 'DATE'
DAYSA   = 871201.,880228.,1.

These ICPs request all dates (00Z time only) in the winter months
(December, January, and February) of 1987-88.

Example 2:
DAYTYPA = 'DATE'
DLDAYSA = 750101.,750130.5,0.5
         , 760101.,760130.5,0.5
         , 770101.,770130.5,0.5

These ICPs request all time samples in January of the three years
1975, 1976, and 1977, assuming two time samples per day.

DEBUG Controls the level of internal diagnostic printout from the Proc-
essor (one CHARACTER value, 'YES' or 'NO', default is DEBUG = 'NO').

'YES' - Enable output of all available diagnostic output.
'NO' - Display only the normal Processor output.

DEFLDcn Explicit field deletion list. Up to 50 pairs of values may be spec-
ified. Each pair consists of the field name (up to 8 CHARACTERS)
followed by a computation type code (INTEGER) (*DFLD). Each spec-
ified field is deleted (i.e., no further processing is done) at the
point immediately following the computation of derived fields (if
any) for the associated computation type. The default is to
continue processing all explicitly requested fields.

NOTES:
1) It is not necessary to request the deletion of fields needed to
compute requested derived fields (*DFLD) if they have been auto-
matically added to the list of fields to process - these fields
are always automatically deleted unless they are explicitly
requested with the <FIELDcn> ICP.
2) Care is necessary when explicitly deleting fields needed to
compute other fields; if a needed field is deleted too soon, a
fatal error will result.

Example:
FIELDA1 = 'T', 'U'
TIMAVGA = 'YES'
SDFLDA1 = 'T', 'SD-T', 'U', 'SD-U'
DEFLDA1 = 'T', 61, 'U', 61

This example requests standard deviations for T and U, then requests that T and U be deleted. Without this explicit deletion request, processing of T and U would continue.

DELREL Controls the disposition of copies of MSS datasets (on the Cray disks) corresponding to the previous input tape when the search for a day moves to a new tape (*MSIN) (one INTEGER value, default is DELREL = 2.)

DELREL = 0 - previous dataset is not removed from the Cray disks.
DELREL = 1 - previous dataset is removed from the $TMPDIR directory.
DELREL = 2 - previous dataset is removed from both the /usr/tmp/ccm and the local ($TMPDIR) directory.

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 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 (*SDFL) for a method of computing some new fields 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).

111 - during input from a hybrid level history tape
11 - during input from a sigma level history tape
12 - during input from a CCMOA history tape
13 - during input from a pressure level history tape
132 - during input from a theta level history tape
14 - during input from a Time Average or Time Series Save tape
20 - after input, before surface-type masking, before spectral
21 - during grid point to spectral transformation
211 - inverse laplacian operator applied in spectral
22 - in spectral space
23 - in spectral space, spectral graphics only
24 - for spectral to grid point transformation
31 - after spectral operations and after surface type masking
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
   bit 3 on - ignored (reserved for constant height surfaces)
   bit 4 on - OK to compute on potential temperature surfaces
   bit 5 on - OK to compute on hybrid coordinate surfaces

For example, a value of 1 means the field can be computed only on
sigma surfaces, a value of 3 (with a bit pattern of 11) means the
field can be computed on either pressure or sigma surfaces, a
value of 16 (with a bit pattern of 10000) means the field can be
only be computed on hybrid surfaces, and a value of 31 means the
field can be computed on any surface. Higher order bits are ignored.

num: 1 2 3 4 5 6 7 8 9 ... 16 ... 31
bits: 1 10 11 100 101 110 111 1000 1001 ... 10000 ... 11111
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 plus four scalar registers. See (*UDFL) for a description of the expression evaluation algorithm.

List of Unary Functions

- .ABS absolute value
- .ALOC natural logarithm
- .ALOG10 common (base 10) logarithm
- .CONST create a constant-value multilevel field
- .DSTIMES delta sigma times each level
- .DSWVSUM delta sigma weighted vertical sum of all levels
- .END expression and definition group terminator
- .EXP e ** operand
- .LEVELnn extract level nn (counting from bottom)
- .MINUS negate operand
- .RANDOM create a pseudo-random (in three-dimensional space plus time) multilevel field with a uniform distribution between 0 and the scalar operand
- .SHIFTDN shift multilevel field values downward 1 level (top two levels of the result are the same, bottom level values are discarded)
- .SHIFTUP shift multilevel field values upward 1 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
:POWER OP1 ** OP2
:LTIMES OP1*(bottom level of OP2) (two fields only)
:UNBLOCK OP1 blocked points given corresponding OP2 values (two fields only)
:LEVMASK zero OP1 levels according to scalar mask; see notes (OP1 field only)
:INVINTB compute a 3-dimensional incremental vertical integral from bottom up; see note 13 (OP2 scalar only)
:INVINTT compute a 3-dimensional incremental vertical integral from top down; see note 14

:MIN minimum(OP1,OP2)
:MAX maximum(OP1,OP2)
:AMOD mod(OP1,OP2)
:EQ .T. if and only if OP1.EQ.OP2
:NE .T. if and only if OP1.NE.OP2
:LT .T. if and only if OP1.LT.OP2
:GT .T. if and only if OP1.GT.OP2
:LE .T. if and only if OP1.LE.OP2
:GE .T. if and only if OP1.GE.OP2
:AND .T. if and only if OP1.AND.OP2 (two fields only)
:OR .T. if and only if 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, so the definition order is important when one derived field is defined in terms of another of the same computation type. 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 and .RANDOM, which generate a field from a scalar constant. Satisfying this requirement for other operators is generally the only reason for using .CONST (computations with scalar operands are more efficient). See example for COSLT below.

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. Note that checking a field for equality with 0. (:EQ) is equivalent to logically negating the field.

5. .SHIFTUP and .SHIFTDN have no effect on single level fields.

6. The :LEVMASK scalar mask must be the second operand, and it must be of type INTEGER. It is interpreted as follows: each bit of the mask corresponds to one level, with the least significant bit mapping to the lowest level; if the mask bit is zero, all values at that level are set to zero, otherwise the value is unchanged. High order bits which do not map to any level are ignored. Although the scalar mask cannot be specified as a binary number, octal values may be input directly - simply append the letter "B" to the integer, as in Cray FORTRAN. For example, the input value 70B may be used to set all levels except 4, 5 and 6 to zero.

7. .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 <LYRSUB>, 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.) See Appendix B for a discussion of vertical integration.

8. All functions except .SHIFTUP, .SHIFTDN, :LEVMASK, and :UNBLOCK check field operands for blocked points (i.e., a field value of 1.E36); if a point in the field operand is blocked, the result is also blocked. :LEVMASK may be used to zero blocked points. The value 1.E36 is not treated as special if it is input as a scalar operand, which means that field values may be set to 1.E36 at a selected set of points, effectively masking those points from subsequent operations. See Appendix B and the example for TSNOW below.

9. .ZAV and .ZAVDEV compute the zonal average independently of any
other zonal averaging operations, and the averages are not qualified or limited in any way by other ICPs. The result of this function is blocked (\texttt{*DEFS}) if, and only if, all points at a given latitude are blocked.

10. Syntax errors encountered during expression evaluation 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 \texttt{.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 \texttt{.END}, the last value preceding the \texttt{.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 (\texttt{*PRNT}), and computations checked by hand.

11. 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 generate error messages as appropriate.

12. 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 \texttt{<FIELDcn>}, but only if all the fields required to compute it are available for that Case. This means that derived field computations cannot combine fields from different Cases. (See \texttt{*CMRG} for a way around this restriction). All computation types can be computed for Cases A and B (subject to input source restrictions for computation types 11-14), but Case C does not exist until Case comparison has been done (\texttt{*ORDR}), so only types 61, 62, 71, and 81 can be computed for Case C.

13. The vertical integral computed by \texttt{:INVINTB} is a multilevel field; each level \( k \) of the result is the sum of the product of the two operands for all levels from the lowest level to level \( k \), i.e.

\[
R = \text{vertical sum from } j=1,k \ (OP1 \ OP2)
\]

where both \( j \) and \( k \) are 1 at the lowest level and increase upward.

14. The vertical integral computed by \texttt{:INVINTT} is a multilevel field; each level \( k \) of the result is the sum of the product of the two operands for all levels from level \( k \) to the highest level, i.e.

\[
R = \text{vertical sum from } j=k,NLVLS \ (OP1 \ OP2)
\]

where both \( j \) and \( k \) are 1 at the lowest level and increase upward, and \( NLVLS \) is the total number of levels for \( OP1 \) or \( OP2 \), whichever is less.
Example:

FIELDA1 = 'WIND', 'TQM', 'TCELS', 'FTS', 'TSNOW', 'ZC-TV', 'COSLT', 'T3', 'Z5', 'PRECmm/d'

These ICPs define twelve derived fields, and requests processing for ten of them. The field TCELS is simply T converted from degrees Kelvin to degrees 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 CCM2 hybrid history tape. QM is the water vapor mass during input from a in each layer CCMOB or CCM1 sigma history tape, 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 is 1. if the surface temperature is between 280. and 300., 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 surface temperature (TSUB1) masked by the snowcover field. The computation 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 the result by 1.E36 results in a "mask" field which is "blocked" at points without snow, and zero at all points with snow. Adding TSUB1 to this field gives TSNOW, which is just TSUB1 blocked at all points without snowcover. Thus, a spatial average of TSNOW is surface temperature averaged over snow only. The 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 (the 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 the bottom to the 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, meaning a leading zero may be necessary. Similarly, Z5 is the fifth level of geopotential height Z2, which is a code-defined derived field computed when reading the input tapes. Note that H1 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 FIELDcn card. Finally, 'PRECmm/d' combines large-scale and convective precipitation, then converts the units to millimeters per day.

DIFFLDn Array of field name triplets used for requesting Case comparison differences (*COMP). All values are up to 8 CHARACTERs each. 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 that no difference fields are computed).

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

These ICPs 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). Setting ORIGFLD = 'NO' will discard 'T' and 'Q' after the the differences are made and continue processing of the difference fields.

DLDAYSc Model times (days) to process specified as an arbitrary number of "do-loop" notation triplets in the form: first day, last day, interval (up to 100 REAL triplets, all nonnegative values). Each do-loop triplet is interpreted exactly as for <DAYSc> "do-loop" notation and is qualified by <DAYTYPc> in the same way. All do-loops are independently expanded, in order, and the resulting list is used as if it had been input for <DAYSc>. This ICP is intended to be used as an alternative to <DAYSc> and is ignored if <DAYSc> is also input.

Example 1:
TIMAVGA = 'YES'
DLDAYSA = 0.,30.,0.5, 365.,395.,0.5, 730.,760.,0.5

II-15
Example 2:
TIMAVGA = 'YES'
DAYTYPA = 'DATE'
DLDAYSA = 810101.,810130.5,0.5
  820101.,820130.5,0.5
  830101.,830130.5,0.5

Example 1 requests that the same 30 days (61 time samples) be averaged for three successive years. Example 2 is similar but requests averaging for the calendar Januaries of 1981, 1982, and 1983.

DPLTCA Specifies the Dicomed camera(s) for the corresponding 'D1' values of DPLTMF (up to 10 CHARACTER values, default is DPLTCA = 10:'FICHE'). Specified values are passed to the sendtg with the CAMERA keyword. The Dicomed is currently configured to produce microfiche for a value of 'FICHE', for 35 mm there are two possible values 'VIEWERBW' for black and white film and 'VIEWERCL' for color film. Do not use the old value of 'FILM' as it no longer works. 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','IS','D1'

Example 2:
DPLTMF = 'D1', 'D1', 'D1', 'IS'
DPLTCA = 'FICHE','VIEWERBL','FICHE'

In Example 1, the default camera is used, resulting in two Dicomed fiche copies and one dispose via rcp. Example 2 requests three Dicomed copies: two on fiche and one on roll film (to the black and white camera), plus a fourth dispose via rcp.

DPLTCP Specifies the filename to which the output plot file will be copied when disposing plot file(s) in conjunction with the 'CP' option of the <DPLTMF> command (one CHARACTER string, up to 72 characters long).

If the last four characters of DPLTCP are ".pid", a unique name will be appended to the filename in place of ".pid". The unique appended name will be of the form ".NE1234-01" where "NE1234" is the job's Cray process id number and "-01" is the current plot dispose group number.

Example:
DPLTMF = 'CP'
DPLTCP = '/crestone/u1/ccmproc2/rje/my.plt.pid'

This ICP requests that 1 copy of the plot file(s) be copied to the crestone file server using the path specified by ICP <DPLTCP>. The plot file will be copied to the local crestone directory.
Note that if the job is submitted via the "qsub" command on shavano, the file's name is used as the process ID number. Thus, using the ".pid" option may not result in a unique filename being generated under this circumstance.

DPLTCT Specifies which Processor case title <TITLEc> is used as the Dicomed plot title if <DPLTIT> is not specified (one CHARACTER value, 'A', 'B', or 'C', default is DPLTCT = 'A'). This ICP is ignored for dispos to <DPLTDF> mainframes other than 'D1'. Maximum lengths 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 <DPLTDF> other than 'D1' (one value, up to 70 CHARACTERS; except for dispos to 'MS' (see below)), the 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 <DPLTDF> values of 'D1'.

For plot file dispos to 'MS', the MSPN is formed as follows: use the Case-independent MSS prefix specified (or defaulted) for <MSPFXO> as the relative directory pathname; for the filename, use the string specified for <DPLTFN> (or the default string 'PLOTS'), followed by two numeric digits giving the plot dispose group number (*PDIS).

Example 1:
DPLTDF = 'UG'
DPLTFN = 'Plots.mcd'

Example 2:
DPLTDF = 'MS'
MSPFXO = '/DIR/SUBDIR/
DPLTFN = 'PIX'

In Example 1, the plot file is returned to the users local system via IRJE with the name 'Plots.mcd'. In Example 2, the MSS name is /DIR/SUBDIR/PIX01 for dispose group 1, /DIR/SUBDIR/PIX02 for dispose group 2, etc.

DPLTIT Specifies the Dicomed plot title (one value, up to 70 CHARACTERS, default is to use the title specified by <DPLTCT>). This ICP is ignored for dispos to <DPLTDF> 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 output hosts or devices for disposing plot files. At plot dispose time, the plot file is disposed for each node specified (*PDIS) (up to 10 CHARACTER values, 2 characters each, default is the single value DPLTMF = 'D1'). The following values generate the dispose conditions as noted below.

Text specified with <DPLTXT> is appended to the TEXT field for all disposes.

'NO' - no plot file dispose (valid only as the first and only value specified)

'IS' - plot file is copied back to any UNIX platform (even off-site) via an rcp command from Shavano. This keyword option requires the use of ICP <DPLTRCPP> to specify the remote path. See the description of <DPLTRCPP>. Synonyms for "IS" include "SA" and "RCP"

'D1' - Dicomed microfilm output device. The following TEXT field is used: 'MDS=nnnnnnn,CAMERA=camera,TITLE=title' where "nnnnnnn" is 1.125 times the plot file size given in 512-byte blocks, "camera" is the string specified or defaulted by <DPLTCA>, and the TITLE keyword is included only if a non-blank title is specified by <DPLTIT> or <DPLTCT> and the corresponding <TITLEc>.

'MP' - the SCD Xerox laser printer (currently driven by the Mass Store Control Processor (MSCP)); the TEXT field includes only the FLNM keyword, and only if a filename is specified with <DPLTFN>.

'MS' - Mass Storage System; the MSPN may be specified with <MSPFXO> and <DPLTFN>. The TEXT field always includes only the FLNM keyword as specified or defaulted by <DPLTFN>.

'CP' - plot file is copied to either somewhere else on shavano or to one of the cross mounted file servers (crestone, meeker etc.) via a cp command from Shavano. This keyword option requires the use of ICP <DPLTCP> to specify the filename. See the description of <DPLTCP>.

'NG' - dispose back to the user through the NCAR machine 'handies'. Only users who have submitted their Processor job through NG should attempt to bring plot files back to their home system via the NG MASNET node. The actual net command used for this node is: netng FLNM=gmeta.nn DF=bi df=bi flnm=gmeta.nn with 01 for the 1st dispose group. Note each time a Processor jobstep disposes the plot file somewhere, the plot dispose
group number is incremented by 1.

'UG' - same as NG but for the UCAR gateway machine windom. The same rules apply as with NG but assume use of the UG by the user.

'SA' - same as 'IS' but assumes the remote machine is SAGE, the ICS sections sun.

'IO','XX','AP','nn',' ' - NO LONGER SUPPORTED

Example:

DPLTMF = 'D1','IS','CP'
DPLTRCP = 'cgd.neit.ucar.edu:/d1/ccmproc2/my.plot'
DPLTCP = '/crestone/u1/ccmproc2/rje/my.plot.pid'

This ICP requests that 3 copies of the plot file(s) be disposed: one to the Dicomed fiche unit, one to the CGD sun using the rcp path specified by ICP <DPLTRCP> and a copy to the crestone file server to the filename specified by the ICP <DPLTCP>. Note that for an rcp copy to work, a correct ".rhosts" file must exist on the remote UNIX machine. See <DPLTCA> and <DPLTFN> for additional examples.

DPLTRCP Specifies the pathname to the remote UNIX platform for disposing plot file(s) in conjunction with the 'IS' option of the <DPLTMF> command.

'pathname' - plot file is copied back to the pathname directory on a remote UNIX machine via an rcp command from Shavano to the CGD suns. This keyword option requires the use of ICP DPLTMF = 'IS' to specify the remote path. See the description of <DPLTMF> below.

If the last four characters of DPLTRCP are ".pid", a unique name will be appended to the filename in place of ".pid". The unique appended name will be of the form "NE1234-01" where "NE1234" is the job's Cray process id number and "-01" is the current plot dispose group number.

Note that if the job is submitted via the "qsub" command on shavano, the file's name is used as the process ID number. Thus, using the ".pid" option may not result in a unique filename being generated under this circumstance.

Example:

DPLTMF = 'IS'
DPLTRCP = 'neit.cgd.ucar.edu:/d1/username/plot1.plt.pid'

This ICP requests that 1 copy of the plot file(s) be disposed to the CGD Sun using the path specified by ICP <DPLTRCP>. The plot file will be copied to the cgdisis Sun directory /dl/username under the name "plot1.plt.NE1234-01"
For the rcp copy to work, the logon name on shavano must match the owner of the directory being written to on the remote machine. Also, a file named ".rhosts" must exist in the user's root directory on the sun. This file is simply a list of the remote machines which have privilege to write files into the user's sun machine. An example of the content .rhost2 file for user USERNAME would be a single line of the form: shavano.ucar.edu username

It is also suggested that the user create a complimentary .rhosts file on shavano to enable the use of remote copy (rcp) and remote shell (rsh) commands.

See <DPLTCA> and <DPLTFN> for additional examples.

This ICP is a synonymn for the ICP DPLTRCP.

Specifies the type of driver code to be used (one CHARACTER value, 'STND' or 'LSD1', default is DRVRTYP = 'STND'). 'STND' indicates that the standard driver for model data processing is to be used. 'LSD1' indicates that the LSD driver is to used; it is assumed that LSD Save Tapes are to be output (*LSDS), so ICPs <SAVLS1c>, <RTLSS1c> and <PWLS1c> MUST also be specified.

Specifies either a case-wise or time-wise ensemble interpretation of the relationship between the [DAYSc] and [TAPESc] lists (one CHARACTER value, 'TIME' or 'CASE', the default is ENSMBLc = 'TIME').

'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.

'TRAGE' - requests ensemble processing (*ENSBI). 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 an ensemble average, i.e., an average of all the members in the ensemble.

List of names of 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 (up to 100 field names, up to 8 CHARACTERS per name).

For a list of names of fields which may be read from Model history tape(s), consult the Users' Guide for the version of the Model that wrote the history tape(s).

The Case suffix can be A, B or C. A "C" suffix is used only when requesting Case C derived fields, since all other Case C fields are
implicitly requested by the ICPs which cause the creation of Case C. 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 that 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','HT1'

This card requests that T, U and V be read from the input tape, and that HT1 (geopotential height) be computed and processed as well. The input tapes must be history tapes because HT1 was requested.

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

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

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

HEMIS = 'BOTH' - plot both Northern and Southern Hemispheres
HEMIS = 'NORTH' - plot Northern Hemisphere only
HEMIS = '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 (up to 8 CHARACTERs), 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 0.0 for all field/levels not specified. The ordering of the triplets with respect to field or level specifier has no significance.

Example:
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 (up to 8 CHARACTERs), 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 0.0, 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 are given a level value of 1000. (regardless of the actual level placement), 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 example causes 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 example 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.
Example 3:

HPCINT = 'U',100.,5., 'U','-1.,10.

This example 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.

HPCNWKI
Size of the INTEGER work array used by CONPAK when creating contour plots (one INTEGER value, default is HPCNWKI = 2000). The default should be quite sufficient.

HPCNWKR
Size of the real work array used by CONPAK when creating contour plots (one INTEGER value, default is HPCNWKR = 5000). The default should be quite sufficient.

HPCONOL
Controls the drawing of continental outlines on all horizontal projection plots (*PHOR) (one CHARACTER value, default is HPCONOL = 'GEOG').

'GEOG' - Geographic continental outlines are drawn. (default)
'MODEL' - Box continental outlines are drawn at Model grid box boundaries between land and either ocean or sea ice grid-points. The Model surface type distribution (the field 'ORO') must be available (*MASK).
'BOTH' - Both geographic and Model continental outlines are drawn as described above.
'NONE' - No continental outlines are drawn.

HPDTCNT
Controls whether or not the continental outlines produced by SUPMAP on horizontal contour plots are solid or dotted (one INTEGER value, the default is HPDTCNT = 0 (solid outlines)). A value of 0 gives solid continental outlines while a value of 1 gives dotted outlines.

HPLFPVn
Array of triplets (all up to 8 CHARACTERs each), listing fields to be plotted as vector arrows on horizontal projections (*PVEC) (up to 33 CHARACTER field name triplets for each keyword, default is no vector fields plotted). The first two values are the names 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].

Example:
FIELDA1 = 'U','V'
HPLFPV1 = 'U','V','WIND'

II-23
This example results in vector arrows for 'WIND', on the horizontal projection requested by <HPVOPT>.

**HPMCINT** Array of 12 values used to specify individual contour intervals on horizontal projection plots. The first quantity in each triplet is the field name (up to 8 CHARACTERS), the second quantity is a level specifier (REAL), the next ten quantities are ten contour values to be plotted. The list is checked for repetition, so if less than 10 contour levels are desired, just repeat the last value such that a total of 10 values are specified. Currently, all levels are plotted.

Example: To plot just the 1 m/s, 5 m/s, and 10 m/s contour levels for all levels of the field 'U':

```
HPMCINT = 'U',1000., 1.,5.,10.,10.,10.,10.,10.,10.10.,10.,10.1
```

**HPPTVAL** Controls whether point value information is printed on horizontal plots (*PVAL) (one CHARACTER value, 'YES' or 'NO', default is HPPTVAL = '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.

**HPRADP** Radius of all horizontal polar projection plots (*PHPO) in degrees of latitude (one REAL value, greater than 0. and less than or equal to 90., default is HPRADP = 70.). The actual radius as a frame fraction is fixed, so this ICP is in effect a 'blowup' option for polar projections. The same radius is used for both polar hemispheres, so that the default of 70., for example, means that the latitude of the outer circumference of the polar plot is 20N for the northern hemisphere and 20S for the southern.

**HPRASPR** Aspect ratio for all horizontal rectangular projection plots (*PHOR) (one REAL value, default is HPRASPR = 1.). This is the number of degrees of latitude to be scaled to the same plot distance as one degree of longitude. Latitude and longitude are always both scaled linearly for this type of plot; specifying a value other than 1.0 for this ICP simply means that they are scaled differently. Also see <HPRSIZE>.

**HPRBNDS** Boundaries for horizontal rectangular plots (cylindrical equidistant projections). (Up to 50 quadruples, all REAL, default is to use the...
global sectioning option <HPRNDIV>.) Each quadruple specifies the
four boundaries of one plot section, in the following order:

1) west (left) border longitude (BLONW)
2) east (right) border longitude (BLONE)
3) south (bottom) border latitude (BLATS)
4) north (top) border latitude (BLATN)

All values are specified in degrees, negative west of Greenwich
and south of the equator. Each quadruple must satisfy the following
constraints:

-90.<=BLATS<=BLATN<=90.
-360.<=BLONW<=180.
-180.<=BLONE<=360.

In addition, the longitude span must not exceed 360. degrees. If
BLONW = BLONE, then the span is assumed to be 360. BLONW may be
greater than BLONE, but east is always to the right. Note that
there are always two ways to specify either border longitude; they
each result in the same plot (any plot may be specified by keeping
both longitudes between plus and minus 180., inclusive). This
option overrides the global sectioning option <HPRNDIV>, and is
ignored unless the ICP <HPRBDIV> is set to 'RECT'. See (*PHRE) for
additional information.

Example:
HPROJ = 'RECT'
HPRNBDIV = -180.,-60.,10.,90.

These two ICPs could be used to produce a limited area plot over
North America.

HPRNBDIV
Number of plot divisions for sectioning horizontal rectangular plots
(cylindrical equidistant projections) (one INTEGER value, default is
HPRNBDIV = 0). The specified value is the number of times the globe
is to be divided (i.e., cut in half), forming individually plotted
sections. In other words, setting HPRNBDIV=n results in the globe
being divided into 2**n plot frames. The first halving is always
along a north-south line, the second is along an east-west line, and
so on, alternating between the two. This keyword is ignored if
HPRNBDIV is specified, and is also ignored unless the ICP <HPRBDIV>
is set to 'RECT'. See (*PHRE) for more information.

Example:
HPROJ = 'RECT'
HPRNBDIV = 3

These two ICPs could be used to produce global plots sectioned
into 8 frames.
Projection option for horizontal plots (*PHOR) (one CHARACTER value, default is HPROJ = '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.

Size of all horizontal rectangular projection plots (*PHOR) relative to the maximum size that can be drawn (one REAL value, default is HPRSIZE = 1.). This should be a fraction between 0. and 1. Since the maximum plot size depends on the aspect ratio set or defaulted with <HPRASPR>, the actual plot size also depends on HPRASPR as well as HPRSIZE. The maximum size of a given plot is determined in the following two-step process: 1) determine the maximum plot span as the larger of: a) the longitude span, and b) the latitude span in degrees longitude (i.e., the latitude span in degrees of latitude divided by the aspect ratio as defined by HPRASPR); 2) divide this maximum plot span by the size fraction as specified by HPRSIZE and scale the plot so that this span is assigned a frame fraction of 0.8. When the global sectioning option is used <HPRNDIV>, the maximum plot size is the maximum of all sections and scaling is the same for all sections. If arbitrary latitude-longitude rectangles are requested with <HPRBNDS>, then each plot is scaled individually.

Array of triplets used to specify the scale factor(s) for horizontal projection (*PHOR) contour plots (*PCON) (CHARACTER, REAL, REAL; the default is HPSCAL = 0., which causes the scale factor to be determined automatically). The first quantity in each triplet is the field name (up to 8 CHARACTERS), the second quantity is a level specifier (REAL), and the third quantity is the scale factor (REAL). Field values are divided 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. The ordering of the triplets with respect to field or level specifier has no significance.

Example:
HPSCAL = 'Q',1000.,1.E-8

This example requests a scale factor of 1.E-8 for the horizontal projection plots for all levels of Q.
HPSMTH

Specifies whether CONPACK is to do a spline smoothing and if it is, specifies the spline tension factor (one REAL value, the default is HPSMTH = 0.0). For this keyword:

- = 0.0 don't do any smoothing of the contours
- < 0.0 do a spline smoothing using the absolute value of the input value as a tension factor. The smoothing is done before any coordinate transformations are performed (i.e. for polar stereographic plots)
- > 0.0 do a spline smoothing using the input value as a tension factor. The smoothing is done after any coordinate transformations are performed (i.e. for polar stereographic plots)

See the description for CONPACK parameter 'T2D' on pg 3-120 of the NCAR Graphics Guide to New Utilities Version 3.00 for more details.

HPSMTHSL

When spline smoothing of contours has been requested this keyword specifies the distance between points used to draw the curves generated by the spline smoother (one REAL value, default value is HPSMTHSL = .01). The value is expressed as a fraction of the window width in the coordinate system in which the smoothing is done. See the description for CONPACK parameter 'SSL' on pg 3--120 of the NCAR Graphics Guide to New Utilities Version 3.00 for more details.

HPVDIN

Vector density grid increment for all horizontal projection vector plots (*PVEC) (one INTEGER value, default is HPVDIN = 1). 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. 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.

HPDIV

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 (up to 8 CHARACTERs), 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 HPDIV = 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'
HPDIV = 'WIND',1000.,30.
These ICPs 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 of the following CHARACTER values, default is HPVOPT = '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 <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 (up to 8 CHARACTERS), 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 HPVSCAL = 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 ICPs 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 arrows
* 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

HYBASGc

Controls processing of hybrid coordinate history tapes labelled as such with the appropriate format type code (one CHARACTER value, 'YES' or 'NO', default is HYBASGc = 'NO'; Cases A and B only).

'NO' - data are assumed to be located on hybrid coordinate surfaces if, and only if, the last decimal digit of the format type code on the header is 4. If the format type code indicates hybrid, then hybrid coefficients must immediately follow the sigma value list on the header. (default)

'YES' - data are processed as if they were located on sigma surfaces even if the format type code's last decimal digit is 4. Any hybrid coefficients following the "sigma" values on the header are ignored.

HYBPRBc

Base pressure to be used in locating hybrid coordinate surfaces when processing hybrid coordinate history tapes <HYBASGc> (one REAL value, expressed in millibars, default is HYBPRBc = 1000, cases A and B only).

ICPECHO

Controls the echoing of ICPs to the disposed plot file(s). The ICPs are always echoed to the print file (one of the following CHARACTER values, the default is ICPECHO = 'PRINT').

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

INDEX

Controls the printing of an index of plot frames in both the print file and in the disposed plot file(s) (one of the following CHARACTER values, the default is INDEX = 'BOTH'). 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>.

INDEX = 'PRINT' - index is written to print file only
INDEX = 'PLOT' - index is written to plot file only
INDEX = 'BOTH' - index is written to both print and plot files
INDEX = 'NO' - no index written

INTDP

Default interpolation type for vertical interpolation to pressure surfaces (*VPRS) (one INTEGER value, default is INTDP = 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.

INTDP = 0 - no interpolation
INTDP = 1 - linear in pressure
INTDP = 2 - linear in natural log of pressure (default).
INTDP = 3 - linear in natural log of natural log of pressure.

INTDT

Default interpolation type code for vertical interpolation to potential temperature surfaces (*VPOT) (one INTEGER value, default is INTDT = 2). This interpolation type is used for all fields not named in the exceptions list <TINTXL>. Specifying a value for this ICP nullifies the default for <TINTXL>. See <TINTXL> 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 INTEGER value, default is LBTDP = 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)
5 - unused
6 - use the ECMWF formulation for below ground points when interpolating to pressure levels. (i.e. extrapolate below ground temperatures down a standard lapse rate, extrapolate below ground geopotential height using the ECMWF formulation and use lowest sigma layers for all other variables; see 'Vertical Interpolation and Truncation of Model-coordinate Data' NCAR/0401/92-33)
LBTDT  Default lower boundary treatment code for vertical interpolation to potential temperature surfaces (*VPOT) (one INTEGER value, default is LBTDT = 4). This lower boundary treatment is used for all fields not named in the exceptions list <TINTXL>. Specifying a value for this ICP nullifies the default for <TINTXL>. See <TINTXL> 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)

LEAPY  Controls whether or not leap years are taken into account in all date computations <DAYTYPc> (one CHARACTER value, 'YES' or 'NO', default is LEAPY = 'NO'). Note that a mismatch between this ICP and the Model's date computations may result in an inappropriate date request when DAYTYPc='DATE' and do-loop notation is used for DAYSc.

'NO' - March 1 is always the next date after February 28. (default)
'YES' - February 29 follows February 28 in leap years.

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, up to 8 CHARACTERs each, default is LMLFPSL = line plots for single level fields only).

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

LXCDIV Array of value pairs used to specify the dividing value(s) for contour line characteristics (*PCON) for all latitudinal cross-section plots (*PLAX) (Up to 200 (CHARACTER,REAL) pairs; default is LXCDIV = 0.0 for any fields not specified by LXCDIV). The first value in each pair is the field name (up to 8 CHARACTERs), and the second is the dividing value (REAL).

Example:

This ICP 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 is the field name (up to 8 CHARACTERs), and the second is the contour interval (REAL). Up to 200 pairs may be specified; the default is LXCINT = 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 ICP requests a contour interval of 5. for T, and a contour interval of 3. for U.

LXLNRNG

Longitude range to be used for all latitudinal cross-section plots (*PLAX). Two REAL longitudes in the following order:
1) west (left) border longitude (BLONW)
2) east (right) border longitude (BLONE)

Both values are specified in degrees, negative west of Greenwich, positive east. Each pair must satisfy the following constraints:

-360.<=BLONW<=180.
-180.<=BLONE<=360.

In addition, the longitude span must not exceed 360. degrees. If BLONW = BLONE, then the span is assumed to be 360. BLONW may be greater than BLONE, but east is always to the right. Note that there are always two ways to specify either border longitude; they each result in the same plot (any plot may be specified by keeping both longitudes between plus and minus 180., inclusive).

Example:
MERAVG = 'YES'
LXLNRNG = -130.,-70.

These two ICPs could be used to produce a latitudinal cross section limited to the Pacific Ocean. Note that specifying the pair of values -130.,290. results in the same plot, while reversing the order to -70.,-130. covers all longitudes EXCEPT over the Pacific.

LXLNSCL

Logarithmic scaling option for the ordinate of all latitudinal cross-section plots (*PLAX) (one CHARACTER value, default is LXLNSCL = '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 Proces-
sor, i.e., logarithmic with respect to the log of the ordinate.

**LXPLOT**

Controls plotting of latitudinal cross sections (*PLAX) at individual longitudes (one CHARACTER value, 'YES' or 'NO', default is LXPLOT = '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 of the following CHARACTER values, default is LXPTVAL = 'NO').

- 'NO' - no point values are printed on latitudinal cross sections
- 'YES' - point values are printed for all requested latitudinal cross-section plots <MERA> <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 factors (*PCON) for all latitudinal cross-section plots (*PLAX). Field values are divided 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 (up to 8 CHARACTERs), and the second is the scale factor (REAL). Up to 200 pairs may be specified; the default is LXSCAL = 0., which causes an appropriate scale factor to be chosen automatically.

Example:

LXSCAL = 'T',10., 'Q',1.E-8

This ICP requests a scale factor of 10. for T and 1.E-8 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) (one REAL value, default is LXSIZE = 1.). 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 <LXASPRIT>.

**LYRSUBc**

List of indices for Model levels to be processed (*LINL) when input is from Model history tapes (Cases A and B only) (up to 100 INTEGER values, default is to process of all Model layers on the history
tape). 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. If exactly two negative values are specified, a range of layers may also be requested by specifying the exactly two negative numbers; they are interpreted as the negative of the lowermost and uppermost layer indices, respectively, in the range (inclusive).

This ICP is independent of the vertical coordinate on which the data are defined and replaces the obsolete ICPs <SIGLEVc>, <SUBPc>, and <SUBTc>.

If the data are defined on sigma or hybrid surfaces, then half-level values are needed in addition to the full level values. If all requested layers are contiguous, then the half-levels are the original layer interfaces for the requested layer subset. If the layers are not contiguous, then the internal layer interfaces are redefined by computing new half-level values as the arithmetic average of adjacent full levels in the subset.

Example 1:
\[ \text{LYRSUBA} = 3,4,6 \]

Example 2:
\[ \text{LYRSUBA} = -2,-6 \]

Example 1 requests that only the third, fourth and sixth levels be processed for Case A. Example 2 requests that only levels 2 through 6 (inclusive) be processed.

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.
MAVG DSP

Disposition of meridional averages, if computed <MERAVG> (one CHARACTER value, default is MAVGDSP = 'PLTDISC').

'PLTDISC' Plot the averages, then discard. The unaveraged data are the basis for all subsequent processing. (Default)

'PLTPROC' Plot the averages, then proceed with processing using the averages as the basis for all subsequent operations. The unaveraged data are discarded.

'PROC' Proceed with processing using the averages as the basis for all subsequent operations. The unaveraged data are discarded. (Same as 'PLTPROC', except that the averages are not plotted.)

This ICP is ignored unless MERAVG = 'YES' is also specified.

MAVGPRN

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

MAVRNG

Range of latitudes for meridional averaging (*MERA), specified as the minimum (southernmost) and maximum (northernmost) value, respectively, in degrees (south latitude is negative, north latitude is positive) (two REAL values, 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 ICPs request meridional averages of all southern hemisphere latitudes.

MBKFR

The fraction of data points in the averaging interval which must be defined (not blocked) (*DEFS) before the meridional average (*MERA) of the points is defined (one REAL value, default is MBKFR = 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.
MEMORY

Number of words of memory to be pre-allocated to the dynamic memory manager at the beginning of the run (one INTEGER value, default is no memory pre-allocated). 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 CHARACTER value, 'YES' or 'NO', default is MERAVG = 'NO').

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

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 INTEGER value, default is MNFRMS = 10000). This ICP is ignored if <DPLTMF> is set to 'NO'.

MSDIRI

Prefix to be used for the local, disk copy of input MSS data files that are Case-independent, and for Case-dependent datasets when a separate prefix is not specified for that Case <MSDIRIC> (one value, up to 70 CHARACTERS, default is MSDIRI = '/usr/tmp/ccm').

Example:
TAPESA = '/USERNAME/TAPEA1'
TAPESB = '/USERNAME/TAPEB1'
MSDIRI = '/usr/tmp/test'

This example requests that the two MSS files /USERNAME/TAPEA1 and /USERNAME/TAPEB1 should both be read with local file names of /usr/tmp/test/USERNAME/TAPEA1 and /usr/tmp/test/USERNAME/TAPEB1 respectively.

MSDIRIC

Prefix to be used for the local, disk copy of input MSS data files. (one value, up to 70 CHARACTERS, default is MSDIRIC = '/usr/tmp/ccm', Case A and B only).

Example:
TAPESA = '/USERNAME/TAPEA1'
MSDIRIA = '/usr/tmp/username'

This example requests that the MSS file /USERNAME/TAPEA1 should be read with a local file name of /usr/tmp/username/USERNAME/TAPEA1.
MSDIRO

Prefix to be used for the local, disk copy of output MSS data files that are Case-independent, and for Case-dependent datasets when a separate prefix is not specified for that Case (one value, up to 70 CHARACTERS, default is MSDIRO = '/usr/tmp/ccm').

Example:
SAVHSTA = '/USERNAME/TAPEA1'
SAVHSTB = '/USERNAME/TAPEB1'
MSDIRO = '/usr/tmp/username'

This example requests that the disk copies of the two MSS files /USERNAME/TAPEA1 and /USERNAME/TAPEB1 should be given the names /usr/tmp/username/USERNAME/TAPEA1 and /usr/tmp/username/USERNAME/TAPEB1 respectively.

MSDIROc

Prefix to be used for the local, disk copy of output MSS data files. (one value, up to 70 CHARACTERS, default is MSDIROc = '/usr/tmp/ccm', Case A and B only)

Example:
SAVHSTA = '/USERNAME/TAPEA1'
SAVHSTB = '/USERNAME/TAPEB1'
MSDIRO = '/usr/tmp/username'

This example requests that the disk copies of the two MSS files /USERNAME/TAPEA1 and /USERNAME/TAPEB1 should be named /usr/tmp/username/USERNAME/TAPEA1 and /usr/tmp/username/USERNAME/TAPEB1 respectively.

MSKAP

Controls how masked area averages are computed (*HORA) (one CHARACTER value, 'YES' or 'NO', default is MSKAP = 'YES'). 'YES' specifies that the field values themselves are to be averaged when masked area averages are requested <MSKFLcn>. '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 CHARACTER value, 'YES' or 'NO', default is MSKAPS = 'NO'). 'YES' specifies that the squares of the field values are to be averaged when masked area averages are requested <MSKFLcn>. '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'.

II-37
MSKAVPR Controls printing of masked area averages computed when time series plots are produced (one CHARACTER value, 'YES' OR 'NO', the default is MSKAVPR = 'NO'). Other masked area averages <MSKFLcn> are always printed.

MSKAZS Controls how masked area averages are computed (*HORA) (one CHARACTER value, 'YES' or 'NO', default is MSKAZS = 'NO'.) 'YES' specifies that the squares of the zonal averages are to be averaged (meridionally) when masked area averages are requested <MSKFLcn>. '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 (up to 8 CHARACTERs),
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)

II-38
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 ICPs 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.

**MSPFXI**

Prefix to be used with a relative MSS pathname <TAPESc> when forming the full MSPN for input datasets that are Case-independent, and for Case-dependent datasets when a separate prefix is not specified for that Case <MSPFXIc> (one value, up to 70 CHARACTERs (79 characters maximum for the full MSPN), default is MSPFXI = ' '). Trailing blanks are removed. See (*MSIN) for additional information.

Example 1:

```
TAPESA = 'TAPEA1','TAPEA2'
TAPESB = 'TAPEB1','TAPEB2'
MSPFXI = '/CSM/CCM1/999/
```

This example requests processing for the following four MSS files:

```
/CSM/CCM1/999/TAPEA1  /CSM/CCM1/999/TAPEA2
```

II-39
Example 2:
TAPESA = 'TAPEA1','TAPEA2'
MSPFXI = 'SUBDIR1/SUBDIR2/SUBDIR3/'

This example requests processing for the following MSS files:

/userid/SUBDIR1/SUBDIR2/SUBDIR3/TAPEA1
/userid/SUBDIR1/SUBDIR2/SUBDIR3/TAPEA2

where "userid" is the UNICOS environment variable $LOGNAME.

MSPFXIc
Prefix to be used with a relative MSS pathname <TAPESc> when forming a full MSPN for input datasets that are Case-dependent (one value, up to 70 CHARACTERS (79 characters maximum for the full MSPN), default is the value specified or defaulted by <MSPFXI>, Case A and B only). Trailing blanks are removed. See (*MSIN) for additional information.

Example:
TAPESA = 'TAPEA1','TAPEA2'
TAPESB = 'TAPEB1','TAPEB2'
MSPFXIA = '/CSM/CCM1/999/'
MSPFXIB = 'MOD/999/'

This example requests processing for the following four MSS files:

/CSM/CCM1/999/TAPEA1 /CSM/CCM1/999/TAPEA2
/userid/MOD/999/TAPEB1 /userid/MOD/999/TAPEB2

where "userid" is determined from the UNICOS environment variable $LOGNAME. Note that using the keyword MSPFXI instead of MSPFXIA in this example would produce identical results.

MSPFXO
Prefix to be used with a relative MSS pathname when forming the full MSPN for output datasets that are Case-independent, and for Case-dependent datasets when a separate prefix is not specified for that Case <MSPFXOc> (one value, up to 70 CHARACTERS (79 characters maximum for the full MSPN), default is MSPFXO = ''). Trailing blanks are removed. See (*MSIN) for additional information.

Example 1:
SAVTAVA = 'STAPEA'
SAVTAVB = 'STAPEB'
MSPFXO = '/MYNAME/SAVTAV/'

This example requests that the following Time Average Save Tapes be written to the MSS:

/MYNAME/SAVTAV/STAPEA /MYNAME/SAVTAV/STAPEB

II-40
Example 2:
SAVTAVA = 'STAPEA'
MSPFXO = 'SUBDIR1/SUBDIR2/SUBDIR3'/

This example requests that the following Time Average Save Tape be written to the MSS:

/userid/SUBDIR1/SUBDIR2/SUBDIR3/STAPEA

where "userid" is determined from the UNICOS environment variable $LOGNAME.

Prefix to be used with a relative MSS pathname <TAPESc> when forming a full MSPN for output datasets that are Case-dependent (one value, up to 70 CHARACTERS (79 characters maximum for the full MSPN), the default is MSPFXOc = the value specified or defaulted by <MSPFXO>). Trailing blanks are removed. See (*MSIN) for additional information.

Example:
SAVTAVA = 'STAPEA'
SAVTAVB = 'STAPEB'
MSPFXOA = '/MYNAME/TAV/999/
MSPFXOB = 'TAV/999/'

This example requests that the following Time Average Save Tapes be written to the MSS:

/MYNAME/TAV/999/STAPEA /userid/TAV/999/STAPEB

where "userid" is the UNICOS environment variable $LOGNAME.

Retention time for all MSS mswrites of output datasets (except plot files) (*PDIS) that are Case-independent, and for Case-dependent datasets when a separate retention time is not specified for that Case with <MSRTOc>. One CHARACTERs value, default is MSRTO = '30'). The MSS interprets this value as a number of days, and imposes additional restrictions on the maximum value (see SCD documentation for the MSS). Currently the maximum retention time is 365 days. Trailing blanks are removed from the string. Plot files are always disposed to the MSS using the system default (i.e., the MSRTO keyword is not used).

Example:
SAVTAVA = '/DIR/STAPEA'
MSRTO = '365'

This example requests that /DIR/STAPEA be disposed to the MSS with a retention time of 365 days.
MSRTOc Retention time for all MSS mswrites of output datasets (except plot files) (*PDIS) that are Case-dependent (one CHARACTER value, default is MSRTOc = '30'). The MSS interprets this value as a number of days and imposes additional restrictions on the maximum value (see SCD documentation for the MSS). Trailing blanks are removed from the string. Plot files are always written to the MSS using the system default (i.e., the RT keyword is not used).

Example:
SAVTAVA = '/DIR/STAPEA'
MSRTOA = '365'

This example requests that /DIR/STAPEA be written to the MSS with a retention time of 365 days.

MSTXTO Character string to be appended to the TEXT field for all MSS mswrites of output datasets (except plot files) (*PDIS) that are Case-independent, and for Case-dependent datasets when a separate TEXT string is not specified for that Case <MSTXTOc> (one value, up to 70 CHARACTERS, default is no additional text.) Trailing blanks are removed from the string. Any valid TEXT keyword(s) or value(s) may be specified, but note that FLNM is always supplied automatically, MVN is automatically added if requested by other ICPs, and the USER keyword is always ignored (because the FLNM keyword is always supplied). The comma separating appended text from previous keywords is always automatically supplied. The double quotes ("" ) delineate the actual text to be passed to the mswrite command.

Example:
SAVTAVA = '/DIR/STAPEA'
MSTXTO = "'PROC02 TIME AVERAGE SAVE TAPE'"

MSRTXOc Character string to be appended to the TEXT field in all MSS mswrites of output datasets (except plot files) (*PDIS) that are Case-dependent. (One value, up to 70 CHARACTERS, default is no additional text, Cases A and B only.) Trailing blanks are removed from the string. Any valid TEXT keyword(s)/value(s) may be specified, but note that FLNM is always supplied automatically, MVN is automatically added if requested by other ICPs, and the USER keyword is always ignored (because the FLNM keyword is always supplied). The comma separating appended text from previous keywords is always automatically supplied. The double quotes ("" ) delineate the actual text to be passed to the mswrite command.

Example:
SAVTAVA = '/DIR/STAPEA'
MSTXTOA = "'PROC02 TIME AVERAGE SAVE TAPE'"

This requests a mswrite to the MSS with the following comment field:

"PROC02 TIME AVERAGE SAVE TAPE"
MXASPRT Specifies the aspect ratio for meridional cross-section plots (*PMEX) (one REAL value, default is MXASPRT = 0.9). 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>.

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 (up to 8 CHARACTERS), and the second is the dividing value (REAL). Up to 200 pairs may be specified; the default is a dividing value of 0.

Example:

This ICP 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 (up to 8 CHARACTERS), and the second is the contour interval (REAL). Up to 200 pairs may be specified; the default is MXCINT = 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 ICP 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 INTEGER value, the default is MXFRMS = 10000). This ICP is ignored if <DPLTMF>='NO'

MXLATRV Controls latitude reversal on meridional cross-sections (*PMEX) (one CHARACTER value, 'YES' or 'NO', default is MXLATRV = '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 (one CHARACTER value, default is MXLNSCL = 'YES'). See (*PMEX).
'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 MXLTRNG = -90.0, 90.0). 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 ICP requests that all meridional cross-section plots cover the northern hemisphere only.

MXPLOT Controls plotting of meridional cross sections (*PMEX) at individual longitudes (one CHARACTER value, 'YES' or 'NO', the default is MXPLOT = '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 (one CHARACTER value, the default is MXPTVAL = 'NO'). See (*PMEX).

'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 divided 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 (up to 8 CHARACTERs), and the second is the scale factor (REAL). Up to 200 pairs may be specified; the default is MXSCAL = 0., which causes an appropriate scale factor to be chosen automatically.

Example:

```
MXSCAL = 'T',10., 'Q',1.E-8
```

This ICP requests a scale factor of 10. for T and 1.E-8 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) (one REAL value, default is MXSIZE = 1.). 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>.

**NDYHSTc**

Maximum number of days to place on a single History Save Tape (one INTEGER value, default is NDYHSTc = 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 INTEGER value, default is NDYTSRc = 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.

**NINTAPc**

Controls the automatic expansion of the <TAPESc> list for the corresponding Case (A or B only). If at least one value is specified for NINTAPc, then exactly one value must be specified for the corresponding TAPESc, and that value must end in a sequence of numeric digits which is to be incremented in order to expand the list. Up to three INTEGER values may be specified for NINTAPc, as follows:

1) N - total number of tapes in the expanded list (N-1 names are
generated). This is a required parameter if NINTAPc is input; there is no default.

2) D - the rightmost D characters in the names are numeric (with leading zeros if necessary) (default is 3 if only one value is input); when a sequence of D nines is exceeded, both d and the length of the name are increased by one character to accommodate the extra numeric digit without overwriting any other part of the name. The value for D specified on input should always be the number of sequence digits in the initial name.

3) I - Numeric increment between successive names (default is 1 if less than three values are input) (D must be input in order to specify I)

If this ICP is not input, then the corresponding TAPESc list is considered complete as input. If an MSS prefix is specified with a separate ICP (*MSIN), it is added after TAPESc list expansion, just as if the expanded list were input.

Example:
MSPFXI = '/CSM/CCM1/239/
TAPESA = 'X23901'
NINTAPA = 3,2

These ICPs are equivalent to the following:
MSPFXI = '/CSM/CCM1/239/
TAPESA = 'X23901','X23902','X23903'

Also note that if a list of 100 tapes were requested, the last would be 'X239100'.

NLCDP Default number of Planetary Boundary Layer levels to copy for vertical interpolation to pressure surfaces (*VPRS) (one INTEGER value, default is NLCDP = 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.

NLCDT Default number of Planetary Boundary Layer levels to copy for vertical interpolation to potential temperature surfaces (*VPOT) (one INTEGER value, default is NLCDT = 0). This number is used for all fields not named in the exceptions list <TINTXL>. Specifying a value for this ICP nullifies the default for <TINTXL>. See <TINTXL> for examples.

NSBDAYc Alternate list of days to process. This ICP may be used instead of <DAYSc> only when <DRVRTYP> is set to 'LSD1' (*LSDS). This ICP is similar to the <DAYSc> ICP, except that the list is two-dimensional with ensemble (*DEFS) as the second dimension. The list of days for each member of the ensemble is terminated with the flag value '.END'; the number of '.END' values then determines the number of
members in the ensemble. For each sublist, the days are specified in the same manner as for \(<\text{DAYSc}\) (i.e., as REAL values). For example,

\[
\text{NSBDAYA} = 0.0, 1.0, 0.5, \text{'END'}, 1.0, 1.5, 2.0, \text{'END'}
\]

specifies the days to process for two members of an ensemble: 0., 1., and 1.5 for the first member, and 1., 1.5, and 2.0 for the second. EACH MEMBER OF THE ENSEMBLE MUST HAVE THE SAME NUMBER OF DAYS. If there is only one ensemble, then the \(<\text{DAYSc}\) ICP may be used instead. The number of members specified for the ensemble must be the same as the number specified for \(<\text{NSBTAPc}\). The total number of values specified must not exceed 100. This ICP is valid for comparison Cases (*COMP) A and B only.

\text{NSBTAPc} Alternate list of full or relative MSPNs for input tapes. This ICP may be used instead of \(<\text{TAPESc}\) only when \(<\text{DRVRTYP}\) is set to 'LSD1' (*LSDS). This ICP is similar to the \(<\text{TAPESc}\) ICP, except that the list is two-dimensional with ensemble as the second dimension. The list of tapes for each member of the ensemble is terminated with the flag value '.END'; the number of '.END' values then determines the number of members in the ensemble. For each sublist, the MSPNs are specified in the same manner as for \(<\text{TAPESc}\) (i.e., as CHARACTER values). For example,

\[
\text{NSBTAPA} = \text{'FILE1A'}, \text{'END'}, \text{'FILE2A'}, \text{'FILE2B'}, \text{'END'}
\]

specifies input tapes for two members of an ensemble: FILE1A for case 1, and FILE2A and FILE2B for case 2. Note that the number of tapes for each member need not be the same. The number of members specified for the ensemble must be the same as the number specified for \(<\text{NSBDAYc}\). If there is only one ensemble, then the \(<\text{TAPESc}\) ICP may be used instead. The total number of characters specified must be less than about 6,000. This ICP is valid for comparison Cases (*COMP) A and B only.

\text{NSDPRNT} Number of decimal digits to consider when printing "significant" differences between Cases A and B (*COMP) (one INTEGER value, the default is no printout of significant differences). Only differences requested with the ICP \(<\text{DIFFLDn}\) 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.

\text{NSVHSTc} Controls the automatic expansion of the \(<\text{SAVHSTc}\) list for the corresponding Case (A, B, or C). If at least one value is specified for \text{NSVHSTc}, then at most one value may be specified for the corresponding SAVHSTc, and that value must end in a sequence of numeric
digits which is to be incremented in order to expand the list. Up to three INTEGER values may be specified for NSVHSTc, as follows:

1) N - Total number of tapes in the expanded list (N-1 names are generated). This is a required parameter if NSVHSTc is input; there is no default.

2) D - The rightmost D characters in the names are numeric (with leading zeros if necessary) (default is 3 if only one value is input); when a sequence of D nines is exceeded, both D and the length of the name are increased by one character to accommodate the extra numeric digit without overwriting any other part of the name. The value for D specified on input should always be the number of sequence digits in the initial name.

3) I - Numeric increment between successive names (default is 1 if less than three values are input) (D must be input in order to specify I)

NSVHSTc is ignored if no values are input for SAVHSTc. If NSVHSTc is not input, then the corresponding SAVHSTc list is considered complete as input. If an MSS prefix is specified with a separate ICP (*MSIN), it is added after SAVHSTc list expansion, just as if the expanded list were input.

Example:
MSPFXO = '/DIR/'
SAVHSTA = 'HST001'
NSVHSTA = 3

These ICPs are equivalent to the following:
MSPFXO = '/DIR/'
SAVHSTA = 'HST001','HST002','HST003'

NSVTSRc Controls the automatic expansion of the <SAVTSRc> list for the corresponding Case (A, B, or C) (up to three INTEGER values, no default). If at least one value is specified for NSVTSRc, then at most one value may be specified for the corresponding SAVTSRc, and that value must end in a sequence of numeric digits which is to be incremented in order to expand the list. The three arguments are:

1) N - Total number of tapes in the expanded list (N-1 names are generated). This is a required parameter if NSVTSRc is input; there is no default.

2) D - The rightmost D characters in the names are numeric (with leading zeros if necessary) (default is 3 if only one value is input); when a sequence of D nines is exceeded, both D and the length of the name are increased by one character to accommodate the extra numeric digit without overwriting any other part of the name. The value for D specified on input should always be the number of sequence digits in the initial name.
NUMPLT

November 1993

ORIGFLD

3) I - Numeric increment between successive names (default is 1 if less than three values are input) (D must be input in order to specify i)

NSVTSRc is ignored if no values are input for SAVTSRc. If NSVTSRc is not input, then the SAVTSRc list is considered complete as input. If an MSS prefix is specified with a separate ICP (*MSIN), it is added after SAVTSRc list expansion, just as if the expanded list were input.

Example:

MSPFXO = '/DIR/'
SAVTSRA = 'TSR001'
NSVTSRA = 3

These ICPs are equivalent to the following:

MSPFXO = '/DIR/'
SAVTSRA = 'TSR001', 'TSR002', 'TSR003'

NUMPLT

Controls the printing of a frame number at the bottom of each plot frame generated (one CHARACTER value, 'YES' or 'NO', default is NUMPLT = 'YES'). 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>.

OFTHSTc

Format for output history save tapes <SAVHSTc> (one CHARACTER value, default is OFTHSTc = 'CCM1'. The CCM2 and CCM1 formats are the 3-record header format, and no longitudinal wrap-around points are written CCMOB is the single-record header format, and two wrap-around points are always written. See the CCM1 and CCM2 Users' Guides for format details.

OFTHSTA = 'CCM2' - Output history tape in CCM2 format.
OFTHSTA = 'CCM1' - Output history tape in CCM1 format.
OFTHSTA = 'CCM0B' - Output history tape in CCM0B format.

Example:

TAPESA = 'CCM1/HT1'
DAYS = 30:-2.
FIELD = 'T', 'U', 'V'
SAVHSTA = 'CCM0B/HT1'
OFTHSTA = 'CCM0B'

This example translates a CCM1 history tape (up to 30 time samples, requested fields only) to the CCM0B format.

ORIGFLD

Controls processing of original fields (Cases A and B) after Case Comparison (*COMP) (one CHARACTER value, 'YES' or 'NO', default is ORIGFLD = 'YES').
'YES' - Continue processing the original fields after comparison.
'NO' - Discard the Case A and B data 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. 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 <NLCDEP> 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) (one INTEGER value, default is PKHSTc = 1 (i.e., no packing)). 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. Allowable range of values is 1 to 4.

PKLS1c Packing density used to pack the data portion of LSD Save Tapes (*LSDS). Exactly one INTEGER value in the range 1 through 4 MUST be specified if, and only if, <DRVRTYP> is set to 'LSD1'. A value of 1 indicates that the data is not to be packed.

PRESSLE List of pressure levels for vertical interpolation from existing sigma or hybrid level data (up to 100 levels in millibars, ordered with largest (closest to ground) values first, REAL, default is no interpolation). The field 'PS' must exist on the input data tape, though it need not be specified in the <FIELDcn> ICP. The specification of at least one value for this keyword is interpreted as a request for interpolation to pressure surface(s).

Note: This ICP cannot be used to request a subset of levels for fields already interpolated to pressure or theta levels. Instead, use <LYRSUBc> to extract specific levels.

Example:

This ICP requests that all fields be interpolated to the 10 pressure levels specified.

PRFLDcn Triplets of field names for time averages of products (*TAVG).

(product = 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 [TIMAVG\textsubscript{c}]. The fields needed to compute the products must be explicitly requested via the [FIELD\textsubscript{cn}] ICP, but the computed product fields should NOT be requested with [FIELD\textsubscript{cn}] (up to 100 CHARACTER field name triplets, default is no computation of products, cases A and B only).

Example:

\begin{verbatim}
FIELD\textsubscript{A1} = 'T','U','V'
TIMAVG = 'YES'
PRFL\textsubscript{A1} = 'T','U','PRO-TU', 'T','V','PRO-TV'
\end{verbatim}

These ICPs request two time average product fields: the product of \( T \) and \( U \), and the product of \( T \) and \( V \), both for Case A in 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 [FIELD\textsubscript{cn}] ICP. This keyword is ignored unless PRINT\textsubscript{c} = 'YES'. Up to 50 field names, up to 8 CHARACTERs each, default is all fields being processed.

**PRINTc** Controls the printing of individual field values for Case c (*PRNT) (one CHARACTER value, 'YES' or 'NO', default is PRINT\textsubscript{c} = 'NO'). See <PRLAT\textsubscript{c}>, <PRLEV\textsubscript{c}>, and <PRLON\textsubscript{c}> before using <PRINTc> = 'YES' since the default values will generate a large amount of printout.

Example:

\begin{verbatim}
PRINTA = 'YES'
PRLONA = -180.
PRLATA = 88.,-88.
PRLEVA = 400.
\end{verbatim}

This prints field values at longitude -180, latitudes 88N and 88S and the vertical level nearest 400 mb or .4 sigma or .4 hybrid (depending on the vertical coordinate being used).

**PRLAT\textsubscript{c}** List of latitudes at which field values are to be printed (*PRNT). This keyword is ignored unless PRINT\textsubscript{c} = '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.

**PRLEV\textsubscript{c}** List of levels at which field values are to be printed (*PRNT), in millibars for pressure, 1000*SIGMA for sigma surfaces, 1000*HYBRID for hybrid surfaces (up to 50 REAL values, in any order, default is all levels being processed). This keyword is ignored unless PRINT\textsubscript{c} = 'YES'. The nearest existing level 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'.

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.

PRNTHD Controls formatted printout of history tape headers (*IHST) for both Cases A and B (one CHARACTER value, default is PRNTHD = 'NO').

'NO' - no printout of input history tape headers (default)
'FULL' - full printout of all headers read, including headers for time samples being skipped
'PART' - same as 'FULL' except that only header scalars are printed

Note: This ICP is ignored for all input types <TYPEc> except for 'CCM0B', 'CCM1' and 'CCM2'; for these history tape formats, however, all vertical coordinate variations (sigma, pressure, potential temperature, and sigma-pressure hybrid) are properly handled.

PTOPc Pressure (in millibars) at the top of the Model, used for computing the code-defined derived field DELPRES (*CDFL) (one REAL value, the default is undefined). See DELPRES description.

PWDHSTc MSS write password, and virtual volume name for all History Save Tapes (*OHST) <SAVHSTc> for the corresponding Case. (One or two values, each up to 8 CHARACTERS). The first value is the write password, which defaults to no password protection. The second value is the virtual volume name, which defaults to the system default virtual volume. The write password must be specified in order to specify a virtual volume; 0 (INTEGER) may be used to indicate no password protection.

Specifying a write password of 'NOMS' will prevent the Save Tape from being disposed to the MSS; it will still be saved on the Cray disk, without a write password.

Example:
PWDTsr

PWDHSTA = 'PASSWD','CTPUBLIC'
PWDHSTA = 'NOMS'

PWDTsr MSS write password, and virtual volume name for all Time Series Save Tapes (*STSR) <SAVTSRc> for the corresponding Case. (One or two values, each up to 8 CHARACTERS). The first value is the write password, which defaults to no password protection. The second value is the virtual volume name, which defaults to the system default virtual volume. The write password must be specified in order to specify a virtual volume; 0 (INTEGER) may be used to indicate no password protection.

Specifying a write password of 'NOMS' will prevent the Save Tape from being disposed to the MSS; it will still be saved on the Cray disk, without a write password.

Example:

PWDTSRA = 'PASSWD','CTPUBLIC'
PWDTSRA = 'NOMS'

PWLS1c MSS write password and virtual volume name (MVN) for all LSD Save Tapes <SAVLS1c> (*LSDS) (one or two CHARACTER values, up to 8 characters each). The first value is the write password, which defaults to no password. The second value is the virtual volume name; the default is determined by the MSS. The write password must be specified in order to specify a virtual volume name; the value 0 may be used to indicate no password protection. Specifying a value of 'NOMS' will cause the LSD Save Tapes to be saved on the Cray disk without being disposed to the MSS.

RATFLDn Array of field name triplets used for requesting Case comparison ratios (*COMP) (up to 100 triplets of names for each keyword, up to 8 CHARACTERS per name, default is no ratio fields computed). 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].

Example:

FIELDA1 = 'T','U'
FIELDB1 = 'T','U'
RATFLD1 = 'T','T-RAT', 'U','U-RAT'

These cards request that ratios be computed (Case A divided by Case B), for both T and U.

II-54
RTLS1c Flag used to determine the order of records written to LSD Save Tapes (*LSDS). Exactly one INTEGER value must be specified if, and only if, <DRVRTYP> is set to 'LSD1'. Given the following ordering of variables,

M - ensemble member,
F - field level,
T - time,

the following flag values result in the indicated ordering:

2 -> F,T,M
4 -> T,M,F
5 -> M,T,F

Negative numbers indicate spectral harmonic coefficients (SHCs) or norms of SHCs. See also <SPCTAVV>,

For example, if RTLS1A = 5, the records vary over the members of the ensemble first, then time, and finally field level. The ensemble members and times are written in the order in which they are specified in <NSBTAPc> and <NSBDAYc>; the field level ordering is more complex (especially when derived fields are computed), but the order is always indicated in the printout.

SAVFHSL Format flag for output Horizontal Slice Save Tapes (all Cases) (*OHor) (one CHARACTER value, default is SAVFHSL = 'NOHEADER'). One of the following values may be specified:

'NOHEADER'  - write horizontal data slices only without any header information. (default)

'HEDER'  - Proceed each horizontal slice with a miniheader. The miniheader contains seven descriptors:

1. Record Number. (integer)
2. Variable Name. (character*8)
3. Vertical Level. (character*8)
4. Number of longitude points. (integer)
5. Number of latitude points. (integer)
6. Day number. (real)
7. Name of this HSL tape. (character*80)

Example:

SAVHSLA = '/DIR/noheader/hsl'

This example requests that data be written to the HSL save tape /DIR/noheader/hsl in the default, 'NOHEADER', format. Assuming that two data slices of the fields 'PS' and 'PHIS' with 129x64 data points were written, the FORTRAN code to read this file would be:

dimension phis(129,64),ps(129,64)
read (11) phis
read (11) ps
Example:
SAVHSLA = '/USERNAME/header/hsl'
SACFHSL = 'HEADER'

This example requests that data with a miniheader be written to /USERNAME/header/hsl. Assuming that this is the same data as in the previous example, the FORTRAN code to read this file would be:

```fortran
real phis(129,64),ps(129,64),day
integer num,lon,lat
character*80 hslname
character*8 field,lev
read (11) num,field,lev,lon,lat,day,hslname, phis
read (11) num,field,lev,lon,lat,day,hslname, ps
```

SAVHSLc MSPN, write password, and virtual volume name for the Horizontal Slice Save Tape (*OHOR) to be written for the corresponding Case (up to 3 CHARACTER values, 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 written to the MSS.

Only the first value need be specified (up to 70 characters, 79 maximum for the full MSPN), in which case there is no password protection and the system default virtual volume is used. If the second and/or third values are specified (up to 8 characters each), 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 'NOMS' will prevent the save tape from being disposed to the MSS; it will still be saved on the Cray disk, without a write password.

Two equivalent examples:

```
SAVHSLA = '/USERNAME/ccm2/300','PASSWD'
```

or

```
SAVHSLA = '/USERNAME/ccm2/300','PASSWD'
MSRTOA = '30'
SHSLZAV = 'NO'
SAVMHSL = 'TSR'
```

SAVHSTc List of MSPNs comprising a set of History Save Tapes (*OHST) to be written (up to 20 MSPNs, up to 70 CHARACTERs each (maximum 79 characters for the full MSPN), 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 MSS. See <PWDHSTc> for specifying write passwords and virtual volumes for these save tapes, and <NDYHSTc> for specifying the maximum number of time samples which may be written to a single tape.

Two Equivalent Examples:

```
MSPFXO = '/USERNAME/ccm2/300/
SAVHSTA = 'HISTP1','HISTP2
PWDHSTA = 'PASSWD'
```

or

```
MSPFXO = '/USERNAME/ccm2/300/
SAVHSTA = 'HISTP1','HISTP2
PWDHSTA = 'PASSWD'
NDYHSTA = 30
MSRTOA = '30'
PKHSTA = 2
BPHSTA = 'NO'
SAVMHST = 'TSR'
OFTHSTA = 'CCM1'
```

SAVLS1c List of MSS full or relative pathnames for output LSD Save Tapes (*LSDS). All values are CHARACTER, up to 70 characters each (maximum 79 for the full MSPN), with an additional maximum of approximately 1400 characters for all values combined. At least one value must be specified if, and only if, <DRVRTYP> is set to 'LSD1'. The output data records are divided as evenly as possible among all specified Save Tapes. The order of the output records is determined by <RTLS1a>.

SAVMHSL Mode flag for output Horizontal Slice Save Tapes (all Cases). (One CHARACTER value, default is SAVMHSL = 'TSR'). One of the following values may be specified:

- 'TSR' - write individual time samples (regardless of the value of <TIMAVGc>) (default)
- 'TAV' - write time average (if, and only if, <TIMAVGc> is 'YES')

Example:

```
SAVHSLA = '/DIR/experiment.42N/SHSL'
SAVMHSL = 'TAV'
TIMAVG = 'YES'
```

This example requests that time averages of the data should be written to the Save Tape: /DIR/experiment.42N/SHSL.
SAVMHST Mode flag for output History Save Tapes (all Cases) (*OHST) (one CHARACTER value, default is SAVMHST = 'TSR'). One of the following values may be specified:

'TSR' - write individual time samples (regardless of the value of \(<TIMAVGc>\) (default)
'TAV' - write time average (if, and only if, \(<TIMAVGc>\) is 'YES')

Example:
SAVHSTA = '/DIR/experiment.43N/SHST'
SAVMHST = 'TAV'
TIMAVGA = 'YES'

This example requests that time averages of the data should be written to the History Save Tape: /DIR/experiment.43N/SHST.

SAVMTSR Mode flag for output of Time Series Save Tapes (all Cases) (*OHOR) (one CHARACTER value, default is SAVMTSR = 'TSR'). One of the following values may be specified:

'TSR' - write individual time samples (regardless of the value of \(<TIMAVGc>\) (default)
'TAV' - write time average (if, and only if, \(<TIMAVGc>\) is 'YES')

Example:
TAPESA = '/CASE1/TAPE1','/CASE2/TAPE1','/CASE3/TAPE1'
DAYSA = 1.,2.,3.,4.,5.
TIMAVGA = 'YES'
ENSMBLA = 'CASE'
SAVTSRA = '/ENSMBLA/SERIES'
SAVMTSR = 'TAV'

This example requests that a time series (5 samples) of ensemble averages (3 cases) be written to /ENSMBLA/SERIES.

SAVTAVc MSPN, write password, and virtual volume name for the Time Average Save Tape (*STAV) to be written for the corresponding Case (up to 3 CHARACTER values, 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 MSS.

Only the first value need be specified (up to 70 characters, maximum 79 characters for the full MSPN), in which case there is no password protection and the system default virtual volume is used. If the second and/or third values are specified (up to 8 characters each), 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 'NOMS' will prevent the Save Tape from being disposed to the MSS; it will still be saved on the Cray disk, without a write password. The keyword [SAVTAPc] is synonymous with [SAVTAVc]; either keyword may be used with the same result.

Example:
SAVTAVA = 'TAVTAP', 'PASSWD'

SAVTSPR

MSPN 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, up to 70 CHARACTERs (79 characters maximum for the full MSPN), 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

MSPN, write password, and virtual volume name for the Time Series Plot Save Tape (*STSP) to be written (up to 3 CHARACTER values, 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 MSS. 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 (up to 70 characters, maximum 79 characters for the full MSPN), in which case there is no password protection, and the Save Tape is written to the system default virtual volume. If the second and/or third values are specified (up to 8 characters each), 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 'NOMS' will prevent the Save Tape from being disposed to the MSS; 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.

Example:
SAVTSPW = 'TSPTAP', 'PASSWD'

SAVTSRc
List of MSPNs comprising a set of Time Series Save Tapes (*STSR) to be written (up to 20 MSPNs, up to 70 CHARACTERs each (maximum 79 characters for the full MSPN), 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 written to the MSS. See <PWDTSRc> for specifying write passwords and virtual volumes 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

SCRUB
Controls the whether existing /usr/tmp/ccm copies of the input MSS dataset should be deleted before acquiring the dataset from the MSS (one CHARACTER value, 'YES' or 'NO', default is SCRUB = 'NO').

'NO' - If the /usr/tmp/ccm copy of the input MSS dataset exists on the Cray disks, use the disk copy as the input dataset. (default).

'YES' - Delete the /usr/tmp/ccm copy of the input MSS dataset and msread a new copy of the dataset from the MSS.

When reading datasets from the MSS, the Processor first looks for a local copy of the dataset under the directory /usr/tmp/ccm. If this copy exists, it is used as the input dataset rather than trying to read it from the MSS. This decreases job turn around time and expense. However, the user has the option of forcing the input dataset to be read directly from the MSS by setting SCRUB = 'YES'.

SDFLDcn
Pairs of field names for time standard deviation computations (*TAVG).

(standard deviation = SQRT(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 ICPs request the computation of standard deviations for both T and U, for Case A, field pass 1.

SFCTCRT
MSPN, write password, and virtual volume name for creating (writing) a Surface Type Save Tape (*SFCT) (up to 3 CHARACTER values, 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 written to the MSS. 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 (up to 70 characters, maximum 79 characters for the full MSPN), in which case there is no password protection, and the save tape is written to the system default virtual volume. If the second and/or third values are specified (up to 8 characters each), 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 'NOMS' will prevent the save tape from being disposed to the MSS; it will still be saved on the Cray disk, without a write password.

Example:
SFCTCRT = 'SFCTAP','PASSWD'

SFCTTAP
MSPN for a Surface Type Save Tape to be used as input (*SFCT) (one value, up to 70 CHARACTERS (maximum 79 characters for the full MSPN), default is no Surface Type Save Tape input). This tape must have been created in a previous run using the ICP keyword <SFCTCRT>.

Two CCM2 T42 Surface Type Save Tapes have been archived on the MSS, /CCMPROC2/ccm2/414/sfct/030101 and /CCMPROC2/ccm2/414/sfct/030701, containing the sea ice distributions for January and July, respectively. The two R15 Surface Type Save Tapes for use with R15 data are: /CCMPROC/SR15JA (January) and /CCMPROC/SR15JL (July).

SHSLZAV
Controls whether zonal averages or point values are output to the Horizontal Slice Save Tape when such output is requested <SAVHSLc> (one CHARACTER value, 'YES' or 'NO', default is SHSLZAV = 'NO').
This zonal averaging is independent of that requested with <ZONAVG>, but the same zonal average qualifiers such as longitude range <ZAVRNG> are used.

SHSLZAV = 'YES' - Only zonal averages are written to the tape.
SHSLZAV = 'NO' - Only point values are written to the 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.

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 CHARACTER value, 'YES' or 'NO', the default is SPCcn = '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).

SPCBPcn List of four spectral truncation parameters used to request spectral bandpass filtering (*SBND) for a particular Case (A or B only) and field pass.

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 bandpass filtering.

CAUTION: All SHC are set to zero outside of a RECTANGLE in (m,n) space; since K>N for a rhomboidal truncation, the maximum value of n (fourth value specified) should be chosen with care. The bandpass rectangle need not be wholly contained within the spectral truncation polygon; the only require-
ment is that SOME values of (m,n) must be common. See (*SPEC) and Model documentation for a discussion of the spectral truncation parameters and the spherical harmonic coefficients (SHC).

Example:

SPCA1 = 'YES'
SPCBPA1 = 1,4,0,30

These ICPs request spectral bandpass filtering for all Case A fields in pass 1. The spherical harmonic coefficients are set to zero for m<1, m>4, and n>30. For data generated with an R15 truncation, this request filters the data zonally, retaining only zonal wavenumbers 1 to 4. The data are then returned to grid point space and processing continues.

SPCDFcn List of fields to be deleted immediately after spectral graphics are completed (up to 100 CHARACTER field names, default is no fields deleted, cases A and B only). This list should include all fields not needed in grid point space.

SPCEFn List of fields to be excluded from spectral processing (up to 100 8-CHARACTER field names, default is no fields excluded, i.e., if spectral processing is requested, all fields being processed are transformed into spectral space, Cases A and B only).

SPCGRAD Defines gradient derived fields for the purpose of computing the horizontal gradient in spectral space (*GRDF). All values are field names (up to 8 CHARACTERS each), organized into 1 to 10 groups with exactly 3 field names in each group. Within each group, the first value is the name of the field whose gradient is desired, and the next two values are the names by which the longitudinal and latitudinal gradient components, respectively, are to be known. The field whose gradient is desired may be an input field or any derived field (*DFLD) computed prior to the transformation into spectral space (*ORDR). This ICP only defines the new gradient derived fields; in order for a gradient component to be computed, it must be explicitly requested with the appropriate [FIELDcn] ICP, and the appropriate [SPCcn] ICP must be set to 'YES'. The field whose gradient is desired need not be explicitly requested with the [FIELDcn] ICP. Each field name must be unique within the entire list, and the gradient component field names must be different than any other derived field names. Although both components of the gradient must be defined (i.e., named), the computation request [FIELDcn] may be limited to one component. Gradients are computed using the same algorithm as the computation of the gradient of the log of surface pressure; see the definitions of the code-defined derived fields DLNPSX and DLNPSY (*CDFL).

Example:

FIELDA1 = 'GRHTLON','GRHTLAT','GRULAT'
SPCA1 = 'YES'
PRESLE = 500.
SPCGRAD = 'HT1','GRHTLON','GRHTLAT','U','GRULON','GRULAT'

This example defines the longitudinal and latitudinal gradients for both geopotential height and eastward wind speed at 500 millibars. Both gradients are computed for geopotential height, but only the latitudinal gradient is computed for the wind. Note that since pressure interpolation is done after spectral operations (*ORDR), the gradient components are computed along sigma surfaces, then interpolated as scalars to 500 millibars. Also note that the spectral transformation has the effect of smoothing HT1 before the gradient is computed.

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 (five INTEGER values, default is no spectral interpolation, i.e., the spectral truncation parameters currently associated with the data are used). Also see (*SPEC) and Model documentation for a discussion of the spectral truncation parameters and spherical harmonic coefficients (SHC). Due to restrictions imposed by the FFT used for spectral transformations, the number of unique longitude points in gridpoint space after spectral interpolation must not have a prime factor larger than 5.

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 ICPs 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.

SPCMNKc List of three spectral truncation parameters (M, N, and K, respectively) used to determine the spectral resolution when transforming data from gridpoint to spectral space (*SPEC) (exactly three INTEGER values, default is to use the truncation parameters currently associated with the gridpoint data, Cases A and B only). This ICP is ignored unless spectral processing is also requested <SPCcn>.

Example:
SPCA1 = 'YES'
SPCMNKA = 24,40,64
SPCTAVV Specifies the final form of data produced by spectral time averages <TIMAVGc> for spectral graphics <SPSNGRF> and/or output to LSD Save Tapes (*OLSD) (one CHARACTER value, default is SPCTAVV = 'NRMSHC').

'NRMSHC' - Time averages of the square of the norm of the SHC are output and/or graphed (default).

'VARSHC' - Time variance of the SHC are output and/or graphed. Nonlinear spectral derived fields (computation type 23) cannot be computed if this value is specified.

'SHC' - Linear Spherical Harmonic Coefficients (SHC) are output. This value is useful only if the SHC are output to LSD Save Tapes (*OLSD) <RTLS1c> and no spectral graphics are requested <SPSNGRF>. Specifying this value when spectral graphics are also requested results in a fatal error.

This ICP is ignored unless both spectral processing <SPCcn> and spectral time averaging have been requested <TIMAVGc>.

SPCVP Defines fields as vector pairs for the purpose of computing vector-pair derived fields in spectral space (*VPDF). 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'
SPCA1 = 'YES'
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', 111, 2, 3, 0, 'U', 'Q', ':TIMES', '.END'
  , 'V*Q', 111, 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> (one REAL value, default is to determine both limits of the ordinate range automatically). The upper limit of the ordinate range is set automatically; the lower limit is the upper limit divided by the value specified for this keyword.

SPSNGRF Controls production of a particular type of graph in spectral space (one CHARACTER value, 'YES' or 'NO', default is SPSNGRF = 'NO'). The graph ordinate is the natural log of the order of the Legendre polynomial and the abscissa is the sum over Fourier wavenumber of the natural log of the SHC times its complex conjugate. See (*SPCR). All fields which exist in spectral space are graphed. This keyword is ignored unless spectral processing is requested <SPCcn>.

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 CHARACTER value, 'YES' or 'NO', default is SURFLEV = '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 (*DFLD) 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 MSPNs for the input data (Case A or B only) (*DATA), ordered with time increasing (up to 300 CHARACTER values, up to 70 characters each, maximum 79 characters for the full MSPN).

Three equivalent examples:
1. TAPESA = '/CSM/ccm2/414/hist/h0001', '/CSM/ccm2/414/hist/h0002', '/CSM/ccm2/414/hist/h0003'
2. MSPFXI = '/CSM/ccm2/414/hist/'
   TAPESA = 'h0001','h0002','h0003',
3. TAPESA = '/CSM/ccm2/414/hist/h0001'
   NINTAPA = 3

These examples each request 3 input tapes: /CSM/ccm2/414/hist/h0001, /CSM/ccm2/414/hist/h0002 and /CSM/ccm2/414/hist/h0003. See the descriptions of the ICPs <MSPFXI>, <MSPFXIc> and <NINTAPc> for more information.

TCFLDcn Triplets of field names for total eddy covariance computations (*TAVG).
(total eddy covariance = x*y*)
   where: stars represent deviations from a zonal average and 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. (up to 100 triplets for each keyword, up to 8 CHARACTERS per name, default is no total eddy covariances computed, cases A and B only)

Example:
FIELDA1 = 'T','U','V'
TIMAVGA = 'YES'
TCFLDA1 = 'T','U','ECV-TU', 'T','V','ECV-TV'

These ICPs request two eddy covariances: the combination of T and U, and also the combination of T and V, both for Case A in field pass 1.

TEMPLEV List of potential temperature levels for vertical interpolation, (up to 100 levels in degrees Kelvin, ordered with smallest (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 from sigma or hybrid surfaces to potential temperature surfaces. The ICP <PRESSLE> cannot be used if values 11-67
are specified for TEMPLEV.

Example:

This example requests that all fields be interpolated to the potential temperature levels specified.

TFWTSc Array of time filtering weights used if, and only if, 'FULL' or 'HALF' is specified for <TIMFILc> (up to about 75 REAL weights). See (*TFIL).

Example:
TIMFILA = 'FULL'
TFWTSA = -1.,1.

These ICPs 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 CHARACTER value, default is TIMAVGc = '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) (one or two CHARACTER values, default is TIMFILc = 'NONE')

first value:

'NONE' - no time filtering is performed (default)

'FULL' - a full set of filter weights must be specified using <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 lowpass 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 bandpass 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 highpass 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 if, and only if, the 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.

TINTMLT
Isentropic vertical interpolation flag indicating how a multiple occurrence (within a vertical column) of a given potential temperature surface is to be handled (one CHARACTER value, default is TINTMLT = 'BLOCK').

'BLOCK' - set all field values to the blocked point value (1.E36) whenever the location of the potential temperature surface is ambiguous. (default)

'TOP' - ignore ambiguity and interpolate to the highest (in elevation) potential temperature surface in the column

TINTXL
Exceptions list for vertical interpolation to potential temperature surfaces (*VPOT). 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 (INTEGER).

The default exceptions list, which is used if, and only if, all three of the ICPs <INTDT>, <LBTDT>, and <NLCDT> 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).

TINTXL = '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:
INTDT = 1
LBTDT = 3
TINTXL = '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 TINTXL are used for Z and MIXRAT.

Example 2:
INTDT = 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.

**TITLEc**

Case description used to identify a particular Case on all plots and printout (up to 64 CHARACTERS, default is TITLEc = ' ' (blanks)). The description can be any sequence of printable characters.

Example:

TITLEA = 'PROCESSOR TEST RUN'

**TITLS1c**

Title to be placed on the data record mini header being written to LSD Save Tapes <SAVLS1c> (*LSDS) (up to 70 CHARACTERS, default is TITLS1c = ' ' (blanks)).

**TSLPASP**

Aspect ratio for all time series line plots (*PTIM) (one REAL value, default is TSLPASP = 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) (one REAL value, default is TSLPSIZ = 1.). Both plot dimensions are affected proportionately. See also <TSLPASP>.

**TSMCASP**

Aspect ratio for all meridional average time series contour plots (*PTMC) (one REAL value, default is TSMCASP = 1.). The aspect ratio is defined as the plot height divided by 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) (one REAL value, default is TSMCSIZ = 1.). Both plot dimensions are affected proportionately. See also <TSMCASP>.

**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 (up to 8 CHARACTERS). 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 distrib-
olution 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:
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.

TSPDYSc Allows renumbering of time samples used to produce all time series plots (*PTIM) (two REAL values, default is to use the day values as requested by <DAYSc> <DAYTYPc>). The first value is the day value to be used for the first time sample; the second value is the day increment to be used between time samples. This ICP is Case-dependent (Cases A, B, and C), but note that when Cases A and B are paired on time series line plots <TSPFNp> the Case A day values are always used. This ICP changes all day values associated with all time series plots produced for the corresponding Case (time axis labels and plot index entries <INDEX>), but all other day values used by the Processor are unaffected.

TSPFNp 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, up to 8 CHARACTERS each). 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 CHARACTER value, default is TSPFPH = 'PAIR').

TSPFPH = 'PAIR' - paired fields are plotted together only (default).
TSPFPH = '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 downward). 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 (up to 8 CHARACTERs). This field must be explicitly requested via the [FIELDcn] ICP.
2) LEVEL value for the level to be plotted - millibars for pressure levels (REAL), 1000.*sigma or 1000.*hybrid for model sigma or hybrid levels. 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 lat-
titude range and minimum fraction of unblocked points for the merid-
ional average computations; otherwise this option is completely
independent of meridional averaging <MERAVG>.

Example:
FIELDA1 = 'V'
TSPMCA1 = 'V',500.,0.,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 auto-
matically, and all negative contour lines are dashed.

TSPMCLR Longitude range to be used for all longitude vs. time (Hovmoller)
plots (*PTMC). Two REAL longitudes in the following order:

1) West (left) border longitude (BLONW)
2) East (right) border longitude (BLONE)

Both values are specified in degrees, negative west of Greenwich,
positive east. Each pair must satisfy the following constraints:

-360.<=BLONW<=180.
-180.<=BLONE<=360.

In addition, the longitude span must not exceed 360. degrees. If
BLONW = BLONE, then the span is assumed to be 360. BLONW may be
greater than BLONE, but east is always to the right. Note that
there are always two ways to specify either border longitude; they
each result in the same plot (any plot may be specified by keeping
both longitudes between plus and minus 180., inclusive).

Example:
TSPMCA1 = 'HT1',500.,100.,0.,0.,5300.
TSPMCLR = -130.,-70.

These two ICPs could be used to produce a Hovmoller plot with the
abscissa limited to the Pacific Ocean. Note that specifying the
pair of values -130.,290. results in the same plot, while reversing
the order to -70.,-130. covers all longitudes EXCEPT over the
Pacific.

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 (up to 8 CHARACTERs). 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, 1000.*hybrid for model hybrid 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.

Example:
FIELDA1 = 'HT1'
TSPPLA1 = 'HT1',500.,82.,-62.,0.,0.

These input cards request a time series plot for the 500. millibar (or .5 sigma or hybrid) geopotential height near Alert, NWT, Canada. The ordinate scale is automatically chosen so that the full range of values is plotted with maximum resolution.

TSPYSCL Sets y (ordinate) scaling for all time series line plots (*PTIM) (one CHARACTER value, default is TSPYSCL = 'LIN')

TSPYSCL = 'LIN' - linear scaling (default)
TSPYSCL = 'LOG' - logarithmic scaling; all field values plotted must be greater than zero or a fatal error will occur when attempting to evaluate the logarithm

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 (up to 8 CHARACTERs). 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, 1000.*hybrid for model hybrid 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> and the ICP ZONAVG = 'YES' should not be called.

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.

TSPZCLR Latitude range for all time vs. latitude plots (*PTZC) (Two REAL latitudes between -90. and +90., default is all latitudes). Both values are specified in degrees, negative for south, positive for north. The minimum of the two values is always used as the southernmost extent of the plot, so the input order is not significant.

Example:
TSPMCA1 = 'HT1',500.,100.,0.,5300.
TSPZCLR = -20.,20.

These two ICPs could be used to produce a time vs. latitude plot limited to latitudes between 20S and 20N.

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 (up to 8 CHARACTERs). 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, 1000.*hybrid for model hybrid 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 REAL value, default is TSZCASP = 1.). The aspect ratio is the plot height divided by the width. See also <TSZCSIZ>.

TSZCSIZ A fraction specifying 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) (one REAL value, default is TSZCSIZ = 1.). Both plot dimensions are affected proportionately. See also <TSZCASP>.

TYPEc Type of tape(s) input for the corresponding <TAPESc> list (one CHARACTER value, default is TYPEc = 'CCM1', case A or B only).

'CCM1' - CCM2, CCM1 or CCM0B format history tape(s). 'CCM0B' is a synonym.
'CCMOA' - Model CCMOA format history tape(s).
'PRESSURE' - Pressure level history tape(s). 'PTP1' is a synonym.
'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).
'THETA' - Potential temperature surface history tape(s). 'TTP1' is a synonym.

NOTE: Both "CCMOA" and "CCM0B" contain the numeral "0", NOT the
When reading History Save Tapes written by the Processor (*OHST), 'CCM1', 'PRESSURE', or 'THETA' should be specified, depending on whether the Save Tape data are on sigma/hybrid, pressure, or potential temperature surfaces, respectively.

UNDEFDF List of code-defined derived fields (*CDFL) to be undefined (up to 100 CHARACTER field names, default is no fields undefined). Undefined fields with this ICP has the effect of freeing up the code defined name so that it can be used for other purposes, for example, defining a user-defined derived field (*UDFL).

VAVGDSP Disposition of vertical averages, if computed <VERAVG> (one CHARACTER value, default is VAVGDSP = 'PROC').

'PLTDISC' - Plot the averages according to <HPROJ>, then discard. The unaveraged data are the basis for all subsequent processing, including plotting according to <HPROJ>.

'PROC' - Proceed with processing using the averages as the basis for all subsequent operations. The unaveraged data are discarded. Averaged data may be plotted later according to <HPROJ>. (Default)

This ICP is ignored unless VERAVG = 'YES' is also specified.

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 or hybrid value (two REAL values, default is to average all levels). All levels which lie within the specified range are averaged. The specified levels need not correspond to actual level values, but a full layer weight is always used (*VERA). This ICP is ignored if no vertical averages are requested.

Example:
VAVRNG = 1000.500.

V BKFR 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 REAL value, default is V BKFR = 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 CHARACTER value, 'YES' or 'NO', default is VERA V = 'NO').

VERAVG = 'YES' - All fields being processed are vertically averaged.
VERAVG = 'NO' - The fields are not vertically averaged.
ZAVGDSP  Disposition of zonal averages, if computed <ZONAVG> (one CHARACTER value, default is ZAVGDSP = 'PLTDISC').

'PLTDISC' - Plot the averages, then discard. The unaveraged data are the basis for all subsequent processing (default).

'PLTPROC' - Plot the averages, then proceed with processing using the averages as the basis for all subsequent operations. The unaveraged data are discarded.

'PROC' - Proceed with processing using the averages as the basis for all subsequent operations. The unaveraged data are discarded. (Same as 'PLTPROC', except that the averages are not plotted.)

This ICP is ignored unless ZONAVG = 'YES' is also specified.

ZAVGPRN  Controls printing of zonal averages (*ZONA) requested with <ZONAVG> (one CHARACTER value, 'YES' or 'NO', default is ZAVGPRN = 'NO').

ZAVGPRN = 'NO' - Zonal averages are not printed (default).
ZAVGPRN = 'YES' - If zonal averages are computed <ZONAVG> then the averages are printed in the output file as well as being plotted.

ZAVRNG  Range of longitudes to be averaged when zonal averaging is requested (*ZONA), specified as the westernmost and easternmost longitudes, respectively, 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.) (two REAL values, default is to average all longitudes).

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.

Example:
ZONAVG = 'YES'
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 REAL value, default is ZBKFR = 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).

\[(\text{zonal covariance}) = [(x) (y)]\]

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 limited to field pass 1.

Example:
- FIELDa1 = 'T','U','V'
- TIMAVG = 'YES'
- ZCVFLA1 = 'T','U','ZCV-TU', 'T','V','ZCV-TV'

This requests 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 CHARACTER value, default is ZONAVG = 'NO').

- ZONAVG = 'NO' - No zonal averages are computed (default).
- ZONAVG = 'YES' - Zonal averages are computed and plotted as meridional cross-sections for all fields in all Cases.

ZORD Controls the addition of a height scale (in km) along the ordinate (outside right) of all vertical cross-section plots (*PMEX) (*PLAX) when hybrid, sigma or pressure is the ordinate (one CHARACTER value, 'YES' or 'NO', default is ZORD = '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. This ICP is ignored when the ordinate is potential temperature.

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 REAL value, default is ZSBKFR = 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.
Pairs of field names used to request time average zonal standard deviation computations (*ZEST).

\[
\text{zonal standard deviation} = \sqrt{(x)(x)}
\]

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 limited to field pass 1.

Example:
FIELDA1 = 'T','U'
TIMAVG = 'YES'
ZCVFIA1 = 'T','ZSD-T', 'V','ZSD-V'

These ICPs request the zonal standard deviations for both T and U for Case A, field pass 1.
This appendix contains a sample for each type of plot that can be produced by the Processor. Also included below each plot are the ICPs used to produce the plot. Copies of the following ICP files are maintained in the shavano directory /ccm/proc/samples/sample.a* where * is the number of the particular sample.

Fig. 1. Temperature contours on all model surfaces (*PHRE) (*PCON). A separate plot is made for each model level for the first day on the input model History Tape (*IHST) using the model continental outline. The plot shown is the first (lowest) level. (A total of 18 horizontal projection plots are created)
**Appendix A**

**November 1993**

**CASE 388**

**CASE 388 T42 plx16**

**Sample plot 2 - Surface type**

C

C INPUT CONTROL PARAMETERS

C

TITLEA = 'Sample plot 2 - Surface type'
TAPESA = '/CCMPROC2/ccm2/414/sfct/030101'
TYPEA = 'SAVTAV'
DAYSA = -1.
FIELDA1 = 'SFCT'
PROJ = 'RECT'
HPPTVAL = 'YES'
HPRNDIV = 3

Fig. 2. Point values for surface type (land, ocean, sea ice) (*PHRE) (*LAHP) (*PVAL). This is the fourth plot in a series of eight. The Surface Type Save Tape (*SFCT) is processed as a Time Average Save Tape (*ITAV). (8 horizontal projection point value plots are created)

A-2
Fig. 3. Geopotential height at 500 millibars contoured on a polar projection (*PHPO) (*PCON). The derived field Z2 (*CDFL) is computed (on model surfaces) from fields on the input history tape, then it is interpolated to 500 millibars (*VPRS) and time averaged over the 31 day period. (1 time averaged polar projection plot is created)
Fig. 4. Horizontal wind vectors on a polar projection (*PHPO) (*PVEC). HPVDIN places vectors at every other data point to avoid overcrowding while HPVSCAL specifies the arrow lengths to be scaled to the magnitude of the vector. (A polar projection vector plot is created)
9.4498-

C
C INPUT CONTROL PARAMETERS
C
TITLEA = 'Sample Plot 5 - Vector wind'
TAPESA = '/CSM/ccm2/414/hist/h0001'
DAYSAA = -1.
FIELDIA = 'U','V'
HPROJ = 'RECT'
HPLFPV1 = 'U','V','WIND'
HPVOPT = 'VECT'
HPVDIR = 2
HPVSCAL = 'WIND',1000.,-1., 'WIND',700.,0., 'WIND',690.,-1.

Fig. 5. Horizontal wind vectors at hybrid level 0.695 on a rectangular projection (*PHRE) (*PVEC). The input history tape contains data on a number of model levels, but plotting for levels greater than 0.7 and less than 0.69 are suppressed by the ICP HPVSCAL. (1 horizontal projection vector plot is created)
Contour from 180 to 300 by 5

C
C INPUT CONTROL PARAMETERS
C
TITLEA = 'Sample Plot 6 - Meridional cross section'
TAPESA = '/CSM/ccm2/414/hist/h0001'
DAYS A = -1.
FIELDA1 = 'T'
ZONAVG = 'YES'
MXCINT = 'T',5.

Fig. 6. Zonally averaged temperature contoured as a meridional cross section (*ZONA) (*PMEX) (*PCON). A meridional cross section plot of all model levels for the first time period on the history tape is made. (1 zonally averaged latitudinal cross section plot is created)

A-6
Sample Plot 7 - Zonally averaged zonal wind.

Day 0.000 U 251.2H

Zonal average from -180.0 to 177.2 degrees

C Input control parameters

C

TITLEA = 'Sample Plot 7 - Zonally averaged zonal wind.'
TAPESA = '/CSM/ccm2/414/hist/h0001'
DAYSA = -1.
FIELDA1 = 'U'
ZONAVG = 'YES'
LMLFPSL = 'U'

Fig. 7. Line plot for zonally averaged zonal wind (*ZONA) (*PLIN). A separate plot is produced for each model level contained on the input history tape. (1 zonally averaged latitudinal cross-section contour plot and 18 line plots are created)
Contour from -18 to 22 by 2

C

C INPUT CONTROL PARAMETERS
C

TITLEA = 'Sample Plot 8 - Latitudinal cross-section'
TAPESA = '/CSM/ccm2/414/hist/h0001'
DAYS A = -1.
FIELDA1 = 'V'
MERAVG = 'YES'
MAVRNG = 30., 60.
LXCINT = 'V', 2.

Fig. 8. Meridionally averaged meridional wind contoured as a latitudinal cross section (*MERA) (*PLAX) (*PCON). All model levels on the input History Tape are contoured. (1 meridionally averaged longitudinal cross section contour plot is created)

A-8
SUM OVER M OF THE NORM SQUARED
CASE414 M,N,K 42 42 42 SPECTRAL TIME AVG.
Sample Plot 9 - Spectral plot

<table>
<thead>
<tr>
<th>1ST DAY</th>
<th>LAST DAY</th>
<th>INC</th>
<th>FIELD</th>
<th>LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>9.000</td>
<td>1.000</td>
<td>UVSQ</td>
<td>992.5H</td>
</tr>
</tbody>
</table>

Fig. 9. Spectral plot of kinetic energy, time averaged in spectral space (*SPGR). The kinetic energy spectra field (UVSQ) is computed from vorticity and divergence, which are automatically derived from U and V on the input history tape (*CDFL). (18 time averaged spectral line plots are created)
Sample Plot 10 - 500-1000 MB layer thickness
CASE 414 CASE 414 p1x19
POINT VALUE OF THICKNES AT LEVEL 1000S
LATITUDE= 9.8 LONGITUDE= -146.2

Fig. 10. Thickness of 500-1000 millibar layer at (9.8N, 146.2.W) plotted as a time series (*PTFL). The geopotential height is computed from fields on the input history tapes (*CDFL), then interpolated to the 1000 and 500 millibar surfaces (*VPRS). The layer thickness is computed as a user-defined derived field (*UDFL). Note the gap on day 10., when the surface pressure fell below 1000mb, resulting a "blocked" value for the undefined 500-1000 millibar thickness. (1 time series line plot is created)
Sample Plot 11 - Temperature transport
CASE 414
ZONAL AVERAGE OF V*T* AT LEVEL 500.OP LATITUDE= 40.5
LONGITUDE RANGE= -180.0 TO 177.2

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 transport is computed as a user-defined derived field (*UDFL). (1 zonally averaged time series line plot is created)
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). (1 masked area average time series paired line plot is created)
Sample Plot 13 - Geopotential height
CASE. 414 CASE 414 p1x19
MERIDIONAL AVERAGE OF Z2 AT LEVEL 500.0P
LATITUDE RANGE = 40.5 TO 60.0

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). (1 meridionaly averaged time series contour plot is created)
C
INPUT CONTROL PARAMETERS
C
TITLEA = 'Sample Plot 14 - Latitude vs time contour plot'
TAPESA = '/CSM/ccm2/414/hist/h0049'
NINTAPA = 4,4
DAYSA = 020101.,020131.,1.
DAYTPA = 'DATE'
FIELDA1 = 'U'
PRESSLE = 500.
TSPZCA1 = 'U',500.,5.,2:0.
ZAVRNG = -60.,0.

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). (1 zonally averaged time series contour plot is created)
Many commonly needed processing capabilities are specifically coded in the Processor as basic options and may be requested in a fairly simple manner. The real power of the Processor, however, lies in the ability to perform complex tasks by combining these basic options in different ways. This is not always straightforward; even for experienced users, the appropriate combinations may not be obvious, and some experimentation and testing are often required. It is possible to perform a wide variety of these complex tasks with relatively few special processing techniques. The main purpose of this appendix is to describe some of these techniques by presenting a series of specific examples which use them. Each example includes a description of a specific problem, a general outline of the technique used to solve it, a listing of all pertinent ICPs, and a discussion of the solution. Although the examples are specific, most of the demonstrated techniques are useful in other situations as well. The ICPs listed here may be copied in whole or part as appropriate.

References to other parts of the documentation are included where appropriate, but familiarity with basic Processor options is assumed, particularly with respect to user-defined derived fields (*UDFL). For a series of introductory examples of the basic Processor options, see the "Introduction to the UNICOS CCM Processor".

B.1 Creating CCM1 Initial Data Tapes

Since the Processor has the ability to output unpacked history tapes (*OHST), it can be used to produce a CCM1 initial data tape, i.e., a history tape containing the initial values for fields required to start the Model. A common application of this capability is changing the CCM's horizontal resolution in the middle of an integration. In this example, a CCM1 history tape is available at a spectral resolution of T42, and it is desired to use this data as the starting point for a T63 Model run. The solution might seem trivial at first: input the T42 history tape to the Processor, extract the fields needed to start the CCM, extrapolate the fields spectrally to T63 (*SINT), then output a history tape. This is indeed the proper basic procedure, but there are a few complications. The main problem is that the initial data tape is the source for most of the CCM's boundary conditions (fields such as surface geopotential and the land-ocean-sea ice flag), and it is not desirable to spectrally extrapolate them. These fields really require special treatment in programs specifically designed to handle them. Suppose, however, that this problem has already been solved for a T63 case (which, in fact, it has) and a T63 history tape containing these boundary fields is available. Since these boundary fields are time invariant in the standard, perpetual date CCM without the hydrological cycle option, they can be obtained from any such T63 history tape. In order to avoid small (but longterm) biases, the tape used should have been written by the CCM in unpacked form. The desired initial data tape can then be created by taking the prognostic fields from the T42 history tape, and the boundary fields from the T63 tape. This can easily be done using the
Processor's Case merge option (*CMRG), but there is another complication: the surface temperature is a prognostic field over land and sea ice, but a climatological boundary field over open ocean. It is possible, however, to construct a user-defined derived field (*UDFL) which combines the surface temperature over non-ocean points from the T42 case, and the ocean temperatures from the T63 case. Here is a complete set of Processor ICPs which solves the entire problem in a single jobstep:

C
C CASE A SPECIFICATION
C
TITLEA = 'T42 HISTORY TAPE CONTAINING PROGNOSTIC FIELDS'
TAPESA = '/CSM/CCM1/244/X244128'
DAYSA = 1275.0
FIELDA1 = 'U','V','T','Q','PS','TSNO'
C
C CASE B SPECIFICATION
C
TITLEB = 'T63 HISTORY TAPE CONTAINING BOUNDARY FIELDS'
TAPESB = '/TMAYER/ANALHT/CCM1/JLT63ST'
DAYSB = 0.0
FIELDB1 = 'TSO','WET','ORO','SNOWH','PHIS'
C
C CASE C SPECIFICATION
C
TITLEC = 'T63 INITIAL DATA TAPE'
FIELDC1 = 'TS'
C
C DATA MANIPULATIONS
C
DERFLD = 'TSO', 11,0,3,0, 'TS' , .END'
   , 'TSNO', 11,0,3,0, 'TS' , .END'
   , 'TS', 61,0,3,0, 'ORO',0.0,':EQ','TSO', ':TIMES'
   , 'ORO',0.0,':GT','TSNO', ':TIMES'
   , ':PLUS', .END'
SPCA1 = 'YES'
SPCINTA = 63,63,63,194,96
ABMERGE = 'YES'
DEFLDC1 = 'TSO',61,'TSNO',61
C
C OUTPUT TAPE SPECIFICATION
C
SAVHSTC = 'T63INIT'
C
C PLOT SPECIFICATIONS
C
HPROJ = 'RECT'
HPPTVAL = 'BOTH'
C
ENDOFDATA -----------------------------------------

Appendix B November 1993 Appendix B

Appendix B November 1993 Appendix B

Appendix B November 1993 Appendix B
In this example, Case A provides the T42 prognostic fields, Case B provides the time-invariant boundary fields, and Case C is the desired initial data. Most of the processing is straightforward: \(<\text{SPCA1}>\) and \(<\text{SPCINTA}>\) request that Case A be spectrally extrapolated to the T63 resolution, and then Case B (already at the T63 resolution), is merged with Case A \(<\text{ABMERGE}>\) to form Case C. The surface temperature, however, is a bit tricky. The user-defined derived fields TSN0 and TSO are simply copies of TS from Cases A and B, respectively. TSN0 is the source for surface temperature for non-ocean gridpoints, while TSO is the source for open ocean gridpoints. It is necessary to make copies of these two fields since they must have unique names if they are to co-exist in Case C. Since the Case A copy (TSNO) must be made before spectral processing, a computation type code of 11 is used \(<\text{DERFLD}>\). The Case C derived field TS is computed after Case merging (computation type code 61) according to the definition specified with \(<\text{DERFLD}>\). The first line in the TS definition results in a field having the TSO values where ORO is equal to zero (ocean), with zeros elsewhere. The second line results in a field having the TSN0 values where ORO is greater than zero (land or sea ice), with zeros elsewhere. Adding these two fields results in the desired surface temperature field TS. TSO and TSN0 are then deleted \(<\text{DEFLDc1}>\) so that they are not written to the output Case C history tape \(<\text{SAVHSTC}>\). (These fields have to be explicitly requested in order to get them into Case C, but because they are explicitly requested, they are not automatically deleted after TS is computed.) Note that the definition for TS includes a test for an exact equality between ORO and 0.0; this does not present a problem here only because the source for ORO is the Case B history tape, which was written in unpacked form. In general, however, such tests should take into account unpacking errors.

There is one additional complication in this example which has been ignored. Fields that are computed by the CCM1 in gridpoint space, such as TS, are not spectrally smooth (with respect to the Model spectral resolution) when they are output to the history tape. In the Processor, transformation from gridpoint to spectral space is an independent step done prior to the extrapolation, and by default, the Model's spectral resolution is used. This has the effect of spectrally smoothing TSN0 to T42 before it is extrapolated to T63, and some of the spatial variation inherent in the gridpoint data is therefore lost. Although this effect is probably not very significant, it may be reduced by explicitly specifying a higher transformation resolution with the ICP \(<\text{SPCMNKc}>\).

**B.2 Unblocking by Vertical Extrapolation**

There are several ways in which the Processor can generate "blocked points", i.e., gridpoints flagged with the special value 1.E36 to indicate missing data. The most common cause of blocked points is vertical interpolation without the option to extrapolate below the lowest field level available (\(<\text{*VERT}>\)). For most fields this kind of extrapolation is undesirable because it is most likely to occur over high ground elevation where the lowest available levels are relatively close together and within the planetary boundary layer. This can produce a steep slope which is unreasonable for extrapolation over long vertical distances. For many operations, however, blocked points are unacceptable (e.g., the spectral analysis options \(<\text{*SPEC}>\)), so some other technique is
required. One solution to the problem is to allow the interpolation to block points, then unblock them using a specified (fixed) vertical profile to extrapolate downward from the lowest unblocked level. If the specified profile is a reasonable one (e.g., one based on climatology), then extrapolations using it are likely to be smoother in the horizontal domain. This technique can be implemented using the user-defined derived field (*UDFL) functions :LEVMASK, .SHIFTDN, and :UNBLOCK, as shown in the following example.

The temperature field T is interpolated to pressure surfaces using the default blocking scheme at the lower boundary. T is then extrapolated downward from the lowest unblocked level in each vertical column using the U.S. Standard Atmosphere lapse rates. The relevant ICPs are first presented in full, followed by a step-by-step description of the computations.

TITLEA = 'sample.b2.sh: Unblocking by Vertical Extrapolation'
TAPESA = '/CSM/ccm2/414/hist/h0001'
DAYSAB = -1.
FIELDA1 = 'TUB5'
PRESSLE = 950., 850., 700., 600., 500., 400., 300., 200., 100.
DERFLD = 'S', 41, 2, 3, 0, 6.0, 'CONST', 1B, ':LEVMASK'
        , 9.0, 'CONST', 2B, ':LEVMASK', ':PLUS'
        , 7.8, 'CONST', 4B, ':LEVMASK', ':PLUS'
        , 8.9, 'CONST', 10B, ':LEVMASK', ':PLUS'
        , 10.5, 'CONST', 20B, ':LEVMASK', ':PLUS'
        , 12.7, 'CONST', 40B, ':LEVMASK', ':PLUS'
        , 10.6, 'CONST', 100B, ':LEVMASK', ':PLUS'
        , 0., 'CONST', 200B, ':LEVMASK', ':PLUS'
        , 0., 'CONST', 400B, ':LEVMASK', ':PLUS', ':END'
        , 'TUB1', 41, 2, 3, 0, 'T', 'T', ':SHIFTDN', 'S', ':PLUS', ':UNBLOCK', ':END'
        , 'TUB2', 41, 2, 3, 0, 'TUB1', 'TUB1', ':SHIFTDN', 'S', ':PLUS', ':UNBLOCK', ':END'
        , 'TUB3', 41, 2, 3, 0, 'TUB2', 'TUB2', ':SHIFTDN', 'S', ':PLUS', ':UNBLOCK', ':END'
        , 'TUB4', 41, 2, 3, 0, 'TUB3', 'TUB3', ':SHIFTDN', 'S', ':PLUS', ':UNBLOCK', ':END'
        , 'TUB5', 41, 2, 3, 0, 'TUB4', 'TUB4', ':SHIFTDN', 'S', ':PLUS', ':UNBLOCK', ':END'
C ENDOFDATA

The desired field is named TUB5 because it is temperature, with up to 5 consecutive levels (counting from the bottom) unblocked by downward vertical extrapolation. First the temperature field T is interpolated to the specified pressure levels. Next, a user-defined derived field S is defined. S contains the temperature "slopes" to be used in the extrapolation; it is horizontally uniform, with a different value at each vertical level. The value at the lowest level prescribes the temperature difference between 950 and 850 millibars; the second lowest level prescribes the difference between 850 and 700, etc. (The values prescribed in this example are approximate values for the U.S. Standard Atmosphere, in units of degrees Kelvin per level, for this particular set of pressure levels.) The definition of S begins by creating a multilevel, constant field with value 6.0. This is then masked with 1B (octal), resulting in a field containing 6.0 at the lowest level, and 0. at all other levels. This is then added to another field containing 9.0 at the second lowest level, with 0. at all other levels. After repeating this for 9 levels, the definition of S is complete. (Note that this definition of S is
specific to the 9 pressure levels requested in this example.) The next field defined is TUB1, which is temperature unblocked at the highest blocked point in every vertical column. The RPN expression begins by pushing a copy of T onto the stack; it will be used later in the expression. A second copy of T is then shifted down by one level, and S is added to it. The result of this subexpression is a field containing values produced by extrapolating downward exactly one level (the extrapolation has been performed at ALL levels simultaneously), and this is exactly what is needed as the second operand for the :UNBLOCK function. The previously stacked copy of T becomes the first operand for :UNBLOCK, and the definition of TUB1 is complete. Note that the extrapolation has been performed in each vertical column independently: the highest blocked level within a given column has been unblocked, regardless of which level it is. TUB2 is next defined in the same manner as TUB1, but starting from TUB1 instead of T. The iteration chain is continued to TUB5, which means that up to 5 levels have been unblocked. The number of iterations is not critical; too many is not a problem, since :UNBLOCK never modifies unblocked values, and too few iterations will be immediately obvious because one or more lower levels will remain blocked. Note that the highest level of S which affects the result is 5, so the definition of S could have been limited to 5 nonzero levels.

Many different extrapolation and interpolation schemes can be implemented using this technique. Note that it is not necessary for the blocked points to be contiguous in the vertical, and also that the blocked point value (1.E36) may be used as a scalar function value (e.g., in a logical test). The key to solving any unblocking problem lies in determining the appropriate definition for the field to be used as the second argument in the :UNBLOCK function. This field need not be fully specified as in this example; it can be tied to other levels of the blocked field, and/or any other available field.

B.3 Vertical Integration with CCM1 data.

The user-defined derived field functions .DSWVSUM, .VSUM, and .DSTIMES <DERFLD> often provide a simple solution to vertical integration problems. For fields located on full sigma levels, .DSWVSUM does a simple integration directly. Minor variations on this scheme can be implemented by operating on the scalar 1. with .DSTIMES to obtain the vertical derivative of sigma, modifying it as desired, and then applying the function .VSUM. (.DSWVSUM is simply the sequential application of .DSTIMES and .VSUM.) This might be useful for integrating over a limited vertical domain, or modifying the integration weights at the integral boundaries. For fields defined on pressure surfaces, the code-defined derived field (*CDFL) DELPRES provides a pressure derivative using ground elevation as a lower boundary; :TIMES and .VSUM can then be used to compute an integral. It is also possible to implement an arbitrary vertical integration scheme by defining a user-defined derived field (*UDFL) containing the desired set of integration weights, and then applying the :TIMES and .VSUM functions. This approach is useful for vertical integration on isentropic surfaces (*VPOT), since there is no simple way to generate the integration weights. The horizontally uniform weight field can be generated in the same manner as the field S in the previous example on unblocking. For weights which are a function of sigma, the weights can be generated manually from the sigma
values which are printed to the output file in any run which reads a history tape. Alternatively, a field of sigma values at full levels may be produced by dividing the code-derived field PRES by the surface pressure PS.

As an example of an alternate integration scheme, consider the following problem. Suppose a new vertical integration scheme for computing geopotential height on full sigma levels is to be evaluated by testing it on a few time samples using the Processor. In order to avoid contamination of the result by interpolation errors, the height field is left on sigma surfaces. The problem is further simplified by computing the new geopotential height HTS-5 on only one sigma surface, say, 0.5. This computation requires a vertical integration from sigma = 1. (level 1) to sigma = 0.5 (level 5 in the standard 12-layer Model) using a vertical array of weights (one column of the matrix called B in the Model documentation). The following ICPs can be used to compute this integral.

```
C
TITLEA = 'CCM1 GEOPOTENTIAL HEIGHT AT SIGMA=.5'
FIELDA1 = 'HTS-5'
DERFLD = 'B' ,11,2,1,0 , .04296089 , '.CONST', 1B , ':LEVMASK',
 , .10022324 , '.CONST', 2B , ':LEVMASK', ':PLUS',
 , .16629604 , '.CONST', 4B , ':LEVMASK', ':PLUS',
 , .24182998 , '.CONST', 10B , ':LEVMASK', ':PLUS',
 , .14183703 , '.CONST', 20B , ':LEVMASK', ':PLUS',
 , ':END',
, 'HTS-5',11,0,1,0 , 'B','TV',':TIMES','.VSUM',287.04 , ':TIMES',
, 'PHIS',':PLUS', 9.8 , ':DIVIDE',':END'
C
ENDOFDATA ----------------------------------
```

In this example, the integration weights are the same as those used in CCM1, and this computation duplicates the .5 sigma level value for the code-defined derived field HT1 (*CDFL). The first step is to set up the integration matrix, B. This is a horizontally uniform field with a specified integration weight at each of the lowest five sigma levels, with zeros at all higher levels. The technique for defining this field is the same as that used in the example on unblocking; see that example's discussion for a detailed explanation of the defining expression. The next step is to write a defining expression for the integral, HTS-5. This is the product of the integration weights B, the virtual temperature TV (which is a code-defined derived field), and the gas constant for air, all integrated using the function .VSUM, followed by the addition of the surface geopotential PHIS, and finally, division by the acceleration due to gravity. The integration algorithm can be modified simply by changing the constants used to define B.

Although it is somewhat cumbersome, it is also possible to define a number of levels of geopotential height using this technique, and then combine the single level fields into a multilevel field. Suppose that HTS-1 through HTS-12 have been defined as above. The following ICPs can then be used to combine them into a multilevel field.
This expression for HTM is similar to the expression for B above in that the same :LEVMASK technique is used to extract specific levels from a number of different multilevel fields and combine them into a single multilevel field. The trick here is to generate the multilevel fields from the previously computed single-level heights HT1, HT2, etc. The function :L1TIMES does this by multiplying each level of the multilevel field ONE (which contains all ones) by a particular single level field. There is an additional complication, however, because a derived field can be defined in terms of at most six different fields. This restriction is circumvented by defining the intermediate fields HTA, HTB, and HTC, and then combining them to get the desired result, HTM.

B.4 Vertical Finite Differences

There are a few capabilities in the Processor which can directly provide vertical finite differences of a vertical coordinate (see the previous example on vertical integration), but finite differencing a field requires a different technique. The key to solving this problem is the user-defined derived field (*UDFL) functions .SHIFTUP and .SHIFTDN, as demonstrated in the following example of a simple, one-sided difference.
This example defines the vertical finite difference of $T$ as the temperature change when going from a given level to the level immediately above. At the highest level, the value for the next lowest level is used. The expression for $\Delta T$ begins by stacking a copy of $T$. A second copy of $T$ is shifted up by one level, then subtracted from the unshifted $T$. The result is then shifted down, completing the definition. It is important to note that when a "out of range" level is shifted onto a defined level, both shift functions simply duplicate the nearest defined level. The second shift in the above expression is taking advantage of this in order to make the top two levels of the result identical. Also note that this expression can be easily modified to make the differencing one-sided downward instead of upward, or to handle the endpoint singularity in a different manner.

The most obvious technique for defining a centered finite difference is to subtract a down-shifted field from its up-shifted counterpart. This technique, however, requires special handling of the top and bottom boundaries if a one-sided difference is to be used at these levels. This special handling can be avoided by taking advantage of the fact that the desired centered difference is simply the arithmetic average of the one-sided upward difference (defined above) and its one-sided downward counterpart. The defining expression can then be written as follows.

C
TITLEA = 'VERTICAL FINITE DIFFERENCE OF T, CENTERED'
FIELDA1 = 'DELTA'
DERFLD = 'DELTA', 111,2,1,0, 'T','T','SHIFTUP',':MINUS','SHIFTDN'

This expression simply continues the previous definition for the one-sided upward finite difference by adding the subexpression for a one-sided downward difference, then averaging the two.

B.5 Time and Space Correlation Statistics

The Processor has simple options for computing a wide variety of statistics which describe the correlation between fields in time and/or space. In particular, the time average (*TAVC) and zonal eddy statistics (*ZEST) options often allow a direct request for the computation of the desired quantity. All of these statistics, however, can also be computed by other, more general techniques. These techniques are described in the following three examples which compute: 1) correlation coefficient, 2) autocorrelation, and 3) covariance of spatial averages.

One of the limitations of the basic option for computing covariances (*TAVC) is that fields must be matched in three-dimensional space. This example shows how to compute the covariance for mismatched vertical levels of two fields, then normalizes the covariance with the product of the standard deviations to obtain a linear correlation coefficient.
The problem of mismatched levels is solved by using the user-defined derived field (*UDFL) function .LEVELnn to extract the desired levels from the multi-level fields after vertical interpolation. Note that the level index in this function refers to the appropriate level, counting up from the surface, AT THE TIME THE FUNCTION IS APPLIED (in this case, AFTER interpolation). Since the resulting fields (T850 and HT500) are defined on only one level, the <CVFLDA1> ICP could have been used to compute the covariance. This example, however, computes the covariance in a user-defined derived field expression instead. The covariance of any two fields can be computed as the time average of the product of the two fields, minus the product of the time averages; the first line in the expression for COR-HT does this using the time averaged product field PROD-THT and the individual time averages T850 and HT500. Note that PROD-THT is defined with the <PRFLDA1> ICP, and this definition requires that both T850 and HT500 be explicitly requested with the <FIELDA1> ICP. The defining expression for the linear correlation coefficient COR-THT continues by dividing the covariance by the product of the standard deviations SD-T and SD-HT. These standard deviations were computed by requesting them with the ICP <SDFLDA1>, but note that they too could have been computed by time averaging the squares of the appropriate fields with <PRFLDA1> and then taking the square roots. Finally, the fields requested with the <PRFLDA1> and <SDFLDA1> ICPs are explicitly deleted by <DEFLDA1>, leaving only the correlation coefficient COR-THT. Note that it has been implicitly assumed here that neither of these correlated fields is constant in time; if this is not the case, then the expression for COR-THT results in a divide by zero.

A similar technique can be used to compute an autocorrelation coefficient for a specified time lag. The lagged and unlagged parts are defined as two separate fields, and the correlation coefficient is computed using the same formula as before. Setting up these two fields, however, is a bit tricky, and requires an additional jobstep. For the sake of variety, the <PRFLDA1> and <SDFLDA1> ICPs are avoided this time, and the problem of a time-invariant field is dealt with by explicitly blocking (*DEFS) the autocorrelation coefficient where necessary.
C
C Jobstep 1 - Merge the Unlagged and Lagged Fields
C
TITLEA = 'sample.b5b.sh: 1. T With lag 0'
TAPESA = '/CSM/cem2/414/hist/h0049'
NINTAPA = 5,2,1
DAYS A = 020101.,020110.1.
DAYTYP A = 'DATE'
FIELDA1 = 'TLO'
T
C
TITLEB = 'sample.b5b.sh: 2. T With lag 2 Days'
TAPESB = '/CSM/cem2/414/hist/h0049'
NINTAPB = 5,2,1
DAYS B = 020102.,020212.,1.
DAYTYPB = 'DATE'
FIELD B1 = 'TL2'
C
DERFLD = 'TLO', 111,2,16,0, 'T', '.END'
, 'TL2', 111,2,16,0, 'T', '.END'
ABMERGE = 'YES'
SAVHSTC = 'test/hist/autocorr'
PWDHSTC = 'NOMS'
ICPECHO = 'BOTH'
DPLTMF = 'NO'
ENDOFDATA

C
C Jobstep 2 - Compute the Autocorrelation Coefficient
C
TITLEA = 'sample.b5b.sh: 2. T Autocorrelation Coeff. for a 2 day lag'
TAPESA = 'test/hist/autocorr'
DAYS A = 10:-2.
FIELDA1 = 'ACC2-T'
TIMAVGA = 'YES'
DERFLD = 'TLO**2',111,2,31,0, 'TLO', 'TLO', ':TIMES', ':END'
, 'TL2**2',111,2,31,0, 'TL2', 'TL2', ':TIMES', ':END'
, 'TLO*TL2',111,2,31,0, 'TLO', 'TL2', ':TIMES', ':END'
, 'COV-T02',61,2,31,0, 'TLO*TL2', 'TLO', 'TL2', ':TIMES', ':MINUS'
, ':END'
, 'P-VAR' ,61,2,31,0, 'TLO**2', 'TLO', 'TLO', ':TIMES', ':MINUS'
, 'TL2**2', 'TL2', ':TIMES', ':MINUS'
, 'TLO*TL2', ':MINUS'
, 'BP-SDV' ,61,2,31,0, 'P-VAR', 'P-VAR', 0., ':LE', 1.E36, ':TIMES'
, ':PLUS', ':SQRT', ':END'
, 'ACC2-T', 61,2,31,0, 'COV-T02', 'BP-SDV', ':DIVIDE', ':END'
SAVHST C = 'AUTOCOR'
C
HPROJ = 'RECT'
ICPECHO = 'BOTH'
HPRBNDS = -180.,0.,-20.,70.
DPLTMF = 'IS'
DPLTRCP = 'neit.cgd.ucar.edu:/d3/ccmproc2/sample.b5b.plt'
ENDOFDATA

B-10
The first jobstep in this example defines the unlagged temperature as Case A, the lagged temperature as Case B, merges (*CMRG) the two fields into Case C, then writes them onto a temporary time series save tape (*STSR). This preliminary jobstep is necessary only because, within a jobstep, case merging is done after time averaging (*ORDR), and the reverse order is required here. The second jobstep performs all of the computations on Case A, which is simply the time series save tape produced in the first jobstep. Note that the days list is the Case A days list from the previous step. The second jobstep first computes the product of the unlagged and lagged temperatures (TL0*TL2), plus their squares (TL0**2 and TL2**2), and then these three fields are time averaged. Next the covariance of TL0 and TL2 is computed (COV-T02) using the formula described in the previous example. In a similar manner, the product of the variances of TL0 and TL2 is computed (P-VAR). This field has a zero value at any gridpoint at which temperature is constant in time, so it cannot be used as a divisor without first blocking those gridpoints. It also could be slightly less than zero due to machine truncation errors, which would cause a problem with the square root needed to convert the variances to standard deviations. The definition for BP-SDV handles both problems by blocking as follows. First a copy of P-VAR is pushed onto the evaluation stack. Then another copy is checked to see if it is less than or equal to zero; if so, the resultant field is 1., otherwise it is 0. This is then multiplied by the blocked point value 1.E36, resulting in a field which is blocked wherever P-VAR is nonpositive, with 0. elsewhere. Adding this to the previously stacked copy of P-VAR then results in a copy of P-VAR which is everywhere positive or blocked. The square root function completes the definition of BP-SDV, which is then divided into COV-T02 to form the autocorrelation coefficient ACC2-T.

Horizontal area averages produced by the simple option (*HORA) can only be plotted or printed; no other operations are possible because the horizontal area averaging option does not produce fields which can be used as input for other operations. In many cases, however, it is possible to produce area averaged fields by successive application of the zonal (*ZONA) and meridional (*MERA) averaging options. This produces the same result as the area averaging option if, and only if, there are no blocked points in the averaging area. The following example uses this technique to compute the covariance of the surface pressure over the southeast Pacific with that over the southwestern Pacific and Indian oceans.

TITLAEA = 'sample.b5c Jobstep 1 - Spatial Covariance - Compute PSSEP'
TAPESA = '/CSM/ccm2/414/hist/h0049'
NINTAPA = 5,2,1
DAYSA = 020101.,020111.,1.
DAYTYPAA = 'DATE'
FIELDA1 = 'PSSEP'
C
ZONAVG = 'YES'
ZAVRNG = -180.,-80.
ZAVGDSP = 'PROC'
MERAVG = 'YES'
MAVRNG = -50.,30.
MAVGDSP = 'PROC'
C

B-11
Due to complications in the ordering of operations (*ORDR), this problem requires four jobsteps. The first jobstep averages PS over the southeast Pacific and writes a time series save tape containing PSSEP. The second jobstep does the same for the southwest Pacific. These two operations cannot
be done as separate Cases in the same jobstep because the spatial averaging operations are not Case-specific. The third jobstep merges the two spatially averaged time series, and the fourth step computes the covariance from the merged time series. These two steps must be separate because time averaging is completed before merging within a jobstep. The resulting covariance is a single number, so it cannot be plotted; the field value printing option (*PRNT) is used to output the result.

B.6 Significance Statistics

Case comparison options (*COMP) allow the direct comparison of two Model Cases, and the computation of differences and/or ratios of fields between the Cases. Because the differences and/or ratios are placed in a third Case (C), however, they cannot be directly combined with the original fields in order to compute derived fields such as significance statistics. The Case Merging option (*CMRG) provides a means around this problem by allowing fields in different Cases to be brought together so that the desired derived fields can be computed. The following example uses this technique to compare a Model experiment with a control, and compute the $t$ significance statistic for the temperature differences.

C
C SPECIFICATION OF CASE A
C
TITLEA = 'sample.b6 - Significance Statistics: Control'
TAPESA = '/CSM/ccm2/414/hist/h0049'
NINTAPA = 5,2,1
DAYSA = 020101.,020105.,1.
DAYTYPA = 'DATE'
TIMAVGA = 'YES'
FIELDA1 = 'TC'
SDFLDA1 = 'TC','SD-TC'
C
C SPECIFICATION OF CASE B
C
TITLEB = 'sample.b6 - Significance Statistics: Control'
TAPESB = '/CSM/ccm2/414/hist/h0159'
NINTAPB = 5,2,1
DAYSB = 050101.,050105.,1.
DAYTYPB = 'DATE'
TIMAVGB = 'YES'
FIELDDB1 = 'TE'
SDFLDB1 = 'TE','SD-TE'
C
C SPECIFICATION OF CASE C
C
ABMERGE = 'YES'
TITLEC = 'MERGED EXPERIMENT AND CONTROL'
FIELDSC1 = 'TDIFF','T-TDIFF'
C
In this example, Case A is a Model control run, and Case B is an experiment for which the temperature changes are to be tested for significance. Five time samples are requested for each Case; the time samples may be instantaneous Model times as assumed here, or each time sample may be a previously computed time average (e.g., the input could be five Time Average Save Tapes for each Case). The time averages and standard deviations of temperature are requested for both Cases. Because the Cases are to be merged, the temperatures are renamed (copied) to TC and TE for the control and experiment, respectively, and the standard deviations are also given different names (SD-TC and SD-TE). After time averaging is completed, the Cases are merged, and the t statistic is computed. This computation is spread out among several temporary fields in order to simplify the algebraic expressions, and also so that the temperature difference TDIFF is available for subsequent processing (e.g., plotting). The computation may also be simplified somewhat by requesting the variances <CVFLDcn> instead of the standard deviations. Note that the sample sizes for Cases A and B have to be hardwired into the computations, and also that the standard deviations <SDFLDcn> computed by the Processor are SAMPLE standard deviations (*TAVG). When plotting the t statistic, contours may be limited to a particular level of significance by applying the .MAX function to the statistic and plotting the resulting new field. Dividing values for contour line attributes (e.g., <HPVDIV>) may also be used to highlight a particular level of significance.

B.7 Event Statistics

It is sometimes useful to determine when and/or where some particular event occurs in a set of data. If an event is defined as one or more fields meeting some specified criteria, then the Processor may be able to recognize and analyze it. For example, a simple event might be defined as "surface temperature below freezing", and one of the statistics which can be computed is the fraction of gridpoints along a latitude line at which this event occurs. The general technique for recognizing such an event is the use of the user-defined derived field (*UDFL) logical functions to flag individual gridpoints as either true (a value of 1.) or false (a value of 0.). A field containing such flag
values can be used to generate two different types of statistics: 1) averaging the flag field gives the fraction of time and/or space at which the event occurs; 2) the flag field can be used as a mask for the computation of statistics based on other field values. The following example demonstrates both uses of an event flag.

Consider the following questions: 1) what is the frequency of occurrence of mostly cloudy skies (total cloud fraction greater than or equal to .8)?, and 2) what is the time average of cloud emissivity? The first question can be answered in a straightforward manner using the technique described above. The second question might appear to be simple at first, but it is actually ill-posed. The problem is that the history tape cloud emissivity field CLOUDE is undefined if there are no clouds. (The field is arbitrarily set to a value of 1, if CLOUD has a value of 0.). A more appropriate question is: When there are clouds, what is their average emissivity? This is a conditional average, i.e., a CLOUDE value is to be included in the average only if CLOUD is nonzero at the corresponding point in time and space. The following ICPs show one way to answer both questions.

C
C Compute: Mostly Cloudy Frequency, Conditional Cloud Emissivity Average
C
TITLEA = 'sample.b7.sh: Mostly Cloudy Freq, Cond. Cloud Emiss Ave'
TAPESA = '/CSM/ccm2/414/hist/h0049'
NINTAPA = 5,2,1
DAYSA = 020101.,020131.,1.
DAYTYP A = 'DATE'
TIMAVGA = 'YES'
FIELDA1 = 'MCSF','CCLDE'
DERFLD = 'MCSF',111,0,31,0, 'TCLD',0.8,':GE',':END'
, 'CLD',111,2,31,0, 'CLOUD',1.E-5,':GT',':END'
, 'CLOUDE',111,2,31,0, 'EFFCLD', 'CLOUD',0.,':EQ'
, 'CLOUD',':PLUS', ':DIVIDE',':END'
, 'CLOUDE',111,2,31,0, 'CLOUDE', 'CLOUD',':TIMES',':END'
, 'CLDB',61,2,31,0, 'CLD',':EQ',1
, 'CLDE',CLDB,':DIVIDE',':END'

The field MCSF is the "mostly cloudy skies frequency". Before time averaging, it is 1 if the total cloud fraction (code-defined derived field TCLD (*CDFL)) is greater than or equal to 0.8, and 0.0 otherwise. After time averaging, MCSF is the desired fractional frequency. Computation of the conditional cloud emissivity average is considerably more complex. First the cloud mask field CLD is computed; this is 1.0 if there are clouds, 0.0 otherwise. In the definition for CLD, note that CLOUD is checked to see if it is greater than or equal to 1.E-5, not 0.0. Although the CCM sets CLOUD to 0.0 if there are no clouds, the history tape packing algorithm modifies these zeros slightly, so it is necessary to consider very small values of CLOUD as a no-cloud condition. The product of CLD and CLOUDE then gives CLDE, which is just the cloud emissivities with zeros where there are no clouds. Both CLD and CLDE are then time averaged, with the no-cloud zeros included in the average. The effect of these zeros in the average for CLDE can be removed by dividing CLDE by CLD, which is simply the frequency of occurrence of clouds. There is
an additional problem, however, because CLD might be 0.0, indicating no clouds at all during the averaging period. In this case the time-averaged emissivity is undefined. This can be handled by dividing instead by CLDB, which is a copy of CLD having the blocked point (*DEFS) value 1.E36 where CLD is zero. The desired statistic is therefore blocked wherever it is undefined. The definition of CLDB begins by pushing a copy of CLD onto the stack. A second copy is checked for equality with 0.0, resulting in a logical field which is 1.0 if CLD is 0.0, 0.0 otherwise. Multiplying this field by the flag value 1.E36 and adding the result to the previously stacked copy of CLD then produces the desired result. The fields produced in this example may then be spatially averaged and/or plotted.

Note that this blocking procedure must be done after the time averaging because a time average is blocked if any of the time samples is blocked. Also note that the time sampling frequency may have an aliasing effect on event statistics such as this one. This can be significant when the sampling frequency is a half-day or more and the fields of interest have a diurnal cycle.

B.8 Complex Time Average Statistics

Some fairly complex time average statistics can be generated by successive applications of the basic time averaging option (*TAVG). If the results of an initial time average step are saved (e.g., on a Time Average Save Tape (*STAV)), then this time average can be used as a single time sample in a subsequent time average. There is no limit to the number of averaging steps which may be performed in this way. This capability is often useful in the analysis of seasonal cycle runs. Suppose, for example, that a monthly average for each of the 12 months in a year is desired, based on output from a 10-year Model run. One way to compute this average is to compute 120 monthly averages first, then average the ten averages for each of the 12 months. The 120 monthly averages can then be the source for interannual variability statistics such as the standard deviation of monthly averages for a given month. An additional step could be a grand 10-year average obtained by averaging the 12 monthly averages.

The 10-year average for a given month, e.g. January, can also be computed more directly. One way is to specify all time samples for the entire set of 31x10 days explicitly. There is, however, a less tedious way. Assuming there are 365 days in one year, and only one time sample per day, the 10-year January average could be requested as follows:

\[
\text{DAYSA}=1. , 30: -2. , 366. , 30: -2. , 731. , 30: -2. , \ldots
\]

with the sequence continuing for the full 10 years. (The prefix n: requests n repetitions of the following value, and -2. is a code interpreted as "the next time sample following the one just processed". Note that if there are two time samples per day, the repetition prefix would be 61.:

When averaging monthly averages to form a yearly average, the differing number of days in each calendar month is probably not worth taking into account. There are, however, other applications where unequal weighting is
necessary. Although the simple time averaging option does not provide for this, it can be done using the time filtering option (*TFIL), as shown in the following example.

Suppose a long Model run has already been averaged in short groups, and a time average save tape is available for each of the following time periods: day 0.5 to 30., day 30.5 to 90., and day 90.5 to 120. A time average for days 0.5 to 120. can be computed from these save tapes, but only if the average for days 30.5 to 90. is weighted twice as much as the other two averages. If the time average save tapes have the names TAV1, TAV2, and TAV3, respectively, then the following ICPs produce the correct overall average.

```
TITLEA = 'UNEQUALLY WEIGHTED TIME AVERAGE'
TAPESA = 'TAV1','TAV2','TAV3'
TYPEA = 'SAVTAV'
FIELDA1 = 'T','U','V'
TIMFILA = 'FULL','NORM'
TFWTSA = 60.,120.,60.
SAVTSRA = 'TAVTOT'
```

Instead of requesting a time average with <TIMAVGA>, these ICPs request a running mean using the time filtering option. Since normalization of the weights has been requested, the weights specified for <TFWTSA> are simply the number of time samples comprising each of the three previously computed averages (here it is assumed that the original sampling frequency was 0.5 days). These ICPs actually request a running mean, but because the number of time samples equals the number of weights, the resulting time series has a length of one, and is in fact the desired time average. Also note that since the Processor regards the result as a time series, the time average is saved on a time series save tape <SAVTSRA>. It cannot be output to a time average save tape because the time averaging option is not being used; it is possible, however, to translate the time series save tape to a time average save tape in a subsequent jobstep. The main disadvantage of using the time filtering option in this way is that all of the time samples must coexist in memory.

B.9. Meridional Streamfunction

The horizontal streamfunction PSI is the only streamfunction defined by the Processor code, but J. Hack and B. Boville have developed a method for computing the zonal mean meridional streamfunction as a user-defined derived field using the new incremental vertical integral function :INVINT. Note that removing the zonal averaging request does NOT result in a three-dimensional streamfunction which is valid at each longitude.

```
```

C-----------------------------------------------
C C Calculation of the zonal mean meridional stream function.
C C J. Hack and B. Boville, January, 1989
C

B-17
For the meridional stream function $\psi$ (kg/s) defined as

\[ v = - \frac{g}{2\pi a \cos(\phi)} \frac{d(\psi)}{dp} \]

\[ \omega = \frac{g}{2\pi (a^2 \cos(\phi))} \frac{d(\psi)}{d(\phi)} \]

where $\phi$ => latitude, $p$ => pressure, and $a$ => earth radius, the zonal mean meridional stream function can be determined from

\[ \frac{\int_0^{p_0} v \, dp'}{p} \]

\[ \psi = \frac{2\pi a \cos(\phi)}{g} \]

which can be approximated by using one of the following sets of ICPs (the integration depends on the vertical coordinate). MPSI is first computed by evaluating the above vertical integral expression. The integration (:INVINTT) is done from the top of the Model down to each half-level using the pressure layer thickness $DP$ (defined at full levels). MPSI is therefore defined at half-levels. The computation is not complete, however, until MPSI is zonally averaged (otherwise the above expression for $\psi$ would need an additional term involving the velocity potential).

The following ICPs compute MPSI for input from history tapes with data on sigma or hybrid surfaces. (DP is code-defined for hybrid coordinate input.)

\[
\]

Must zonally average to complete computation

\[
\text{ZONAVG} = 'YES'
\]

The following ICPs compute MPSI for data on pressure surfaces, regardless of the type of input. Note the use of the ICP PTOA to control the computation of the (code-defined) pressure layer thickness DELPRES for the highest layer. Alternatively, the code-defined derived field PRESP could be used to compute the pressure layer thickness with a different algorithm.

\[
\text{PTOA} = 0.
\text{DERFLD} = 'MPSI', 41, 1, 31, 0, 'V', 'DELPRES', 40031556., ':TIMES', '.TCOSL'
\]
B.10 Isentropic Potential Vorticity

The following set of ICPs for computing "isentropic potential vorticity" was developed by P. Rasch and B. Boville. Although the ICP TEMPLEV requests interpolation to potential temperature surfaces, what interpolation, if any, is performed is optional since the computation of PV is complete before any interpolation is begun. The definition of PV is quite general, depending only on the availability of U, V, THETA, and PRES. (THETA and PRES are code-defined derived fields which can be computed on either sigma or hybrid input surfaces.) Note that spectral processing must be requested in order to compute VOR and the horizontal gradients of THETA, and that since THETA is spectrally smoothed in the process DELTH should be computed afterward for consistency. PRES is excluded from spectral processing simply because it is not needed in spectral space.

C-------------------------------
C ICPs for generating plots of potential vorticity on C "isentropic" surfaces. The actual quantity computed is Ertel's C potential vorticity using potential temperature as the invariant C quantity. This is often referred to as isentropic potential vorticity C or "IPV". The field name here is "PV".  C PV = -g Zeta d(theta)/dp, where Zeta is the absolute vorticity on theta C surfaces.
C Zeta is approximated by a standard coordinate transformation from the C input surfaces to theta surfaces.
C PV = -g[(vor+f) d(theta)/dp - [d(theta)/dx dv/dp - d(theta)/dy du/dp]] C where vor, d(theta)/dx, and d(theta)/dy are on the input surfaces and C are obtained through spectral transforms.
C "vor" is the relative vorticity, a code defined derived field.
C C The -g factor is included to obtain the result in commonly used units C K m^-2 kg / s.
C C-------------------------------
C C fields to process, THETA required for interpolation to THETA surfaces
C
FIELDA1 = 'PV', 'THETA'
C
C output surfaces for potential vorticity
C
TEMPLEV = 400.,428.
C
C spectral transform control variables,
C
SPCA1 = 'YES'
SPCEFA1 = 'PRES'
SPCCGRAD = 'THETA','THX', 'THY'
C
C Define fields necessary to compute Ertel's potential vorticity using
C potential temperature as the invariant.
C DELTH - delta theta - d(theta)/dp = DELTH/DELP
C DELU - delta u
C DELV - delta v
C PV1 - d(theta)/dp (zeta+f), where zeta is relative vorticity on
C the input surfaces
C PV2 - k cross grad(theta) dot d(V)/dp, coordinate transformation
C to convert relative vorticity term to
C potential temperature surfaces
C PV - Ertel's potential vorticity using theta as the invariant

DERFLD = 'DELTH',31,2,31,0
 , 'THETA','THETA','SHIFTUP','MINUS'
 , 'THETA','THETA','SHIFTDN','MINUS'
 , 'MINUS','END'
 , 'DELU',31,2,31,0
 , 'U','U','SHIFTUP','MINUS'
 , 'U','U','SHIFTDN','MINUS'
 , 'MINUS','END'
 , 'DELV',31,2,31,0
 , 'V','V','SHIFTUP','MINUS'
 , 'V','V','SHIFTDN','MINUS'
 , 'MINUS','END'
 , 'DELP',31,2,31,0
 , 'PRES','PRES','SHIFTUP','MINUS'
 , 'PRES','PRES','SHIFTDN','MINUS'
 , 'MINUS','END'
 , 'PV1',31,2,31,0
 , 1.458E-04,'CONST','TSINL','VOR','PLUS'
 , 'DELTH','TIMES','DELP','DIVIDE','-9.81','TIMES','END'
 , 'PV2',31,2,31,0
 , 'THX','DELV','TIMES'
 , 'THY','DELU','TIMES','MINUS'
 , 'DELP','DIVIDE','-9.81','TIMES','END'
 , 'PV',31,2,31,0
 , 'PV1','PV2','MINUS','END'

C
C map control variables, 1.0E6 is normally a good scaling factor for PV
C
HPSCAL = 'PV',1000.,1.0E+6
HPROJ = 'POLAR'
B.11 FORTRAN code for reading CCM2 history tapes.

The following code will read a CCM2 history tape. A copy of this code is stored at NCAR as /ccm/proc/samples/ezread.sh

Program ezread

C Example code to read in one level of data from a ccm2 history tape.
call msread (ier,'fort.11','/CSM/ccm2/414/hist/hOO01',' ',' ') FIELD = "T" ! Plot the TOP level of this field.

do 40 ndays =1, 1 ! Loop over days
C Read the integer, character and real headers (these are never packed).
read(11,iostat=io,end=96,err=97) ih(1),(ih(k),k=2,ih( 1))
read(11,iostat=io,end=96,err=97) (ch(k),k=1,ih(31))
read(11 ,iostat=io,end=96,err=97) (rh(k),k=1,ih(32))

C Print the integer, character and real header values
write (6,'(10i9)') (ih(i),i=1,37)
write (6,'(13a8)') (ch(i),i=1,11)
write (6,'(13a8)') (ch(i),i=12,ih(37)-1)
write (6,*) 'N Field Units Lev Point Pack'
write (6,'(i10,2a9,3i6)') (i+1,(ch(ih(37)+2*i+k),k=O,1), (ih(ih(36)+3*i+k),k=0,2),i=0,ih(16)-1)
write (6,*) 'Levels:'
write (6,'(10f10.4))') (rh(i),i=ih(33),ih(34)-1)
write (6,*) 'Latitudes:'
write (6,'(10f10.4))') (rh(i),i=ih(34),ih(35)-1)
C Read the data records

do 30 lat=1,ih(11) ! Loop over latitudes
  if (iflds(3,1) .gt. 1 ) then ! If data is packed....
    read (11,iostat=io,end=98,err=99) xlat,xlon,
    istart=1 ! Packed data pointer
    kslice=1 ! Unpacked data pointer
    ndens = iflds(3,1) ! Find packing density
    do 20 num=1,ih(16) ! Loop over fields
      ksize = 2+((ih(9)+ndens-1) / ndens) ! Len of 1 packed longitude
      levs = ih(12) ! Find number of levels
      if (mod(iflds(1,num),10).gt.1) levs = 1
      if (lat.eq.1 and.(cflds(1,num).eq.FIELD))
        istar=kslice ! pointer to FIELD
      do 10 lev=1,levs ! loop over levels
        call unpka (slice(kslice), ih(9),
            ! Unpack the data from
        buffer(istart), ndens) ! buffer into slice
        kslice = kslice + ih(9) ! Advance unpacked pointer
        istart = istart + ksize ! Advance packed pointer
        10 continue
      20 continue
  else ! If data is not packed....
    if (lat.eq.1) then 
      do num=1,ih(16)
        if(cflds(1,num).eq.FIELD) istar=iflds(2,num)
      enddo
    endif
    read (11,iostat=io,end=98,err=99) xlat,xlon,
    (slice(len),len=1,ih(7)-2) ! Read unpacked data slice
  endif
  if (istar.eq.0) then
    print *, FIELD,' was not found on this tape. Stopping...' 
    stop 999
  endif
  do lon=istar,istar+ih(10)-1 ! loop over longitude
    data(lon-istar+1,int(xlat))=slice(lon) ! copy FIELD to array data
  enddo
  30 continue

do lat=1,ih(11) ! Loop over latitudes
  write (6,'(i3,f9.4,9(lpe12.4))') ! print some data
  lat, rh(ih(34)+lat-1),(data(lon,lat),lon=1,8)
  enddo
40 continue

96 write (6,*) 'EOF in rdhdr from unit ',11,' code=',io 
write (6,'(10i9)') (ih(i),i=1,37)
goto 999
97 write (6,*), 'ERR in rdhdr from unit ',11,' code=',io
write (6,'(10i9)') (ih(i),i=1,37)
goto 999
98 write (6,*), 'EOF in rdata from unit ',11,' code=',io
goto 999
99 write (6,*), 'EOF in rdata from unit ',11,' code=',io
999 stop
end
<table>
<thead>
<tr>
<th>Index</th>
<th>November 1993</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABMERGE (ICP)</td>
<td>I-57, 87, II-1, B-2, 3, 10, 12, 13</td>
<td>ABMERGE (ICP)</td>
</tr>
<tr>
<td>ABS (DERFLD operator)</td>
<td>II-10</td>
<td>ABS (DERFLD operator)</td>
</tr>
<tr>
<td>ALOG (DERFLD operator)</td>
<td>II-10</td>
<td>ALOG (DERFLD operator)</td>
</tr>
<tr>
<td>ALOG10 (DERFLD operator)</td>
<td>II-10</td>
<td>ALOG10 (DERFLD operator)</td>
</tr>
<tr>
<td>:AMOD (DERFLD operator)</td>
<td>II-11</td>
<td>:AMOD (DERFLD operator)</td>
</tr>
<tr>
<td>:AND (DERFLD operator)</td>
<td>I-45, II-11, 12, 14</td>
<td>:AND (DERFLD operator)</td>
</tr>
<tr>
<td>Area average</td>
<td>I-26, 44, 92, 93, 102, 104, 105, II-37:39, 71, 73, A-12, B-11</td>
<td>Area average</td>
</tr>
<tr>
<td>BPHSTc (ICP)</td>
<td>I-41, 42, II-1, 1, 57</td>
<td>BPHSTc (ICP)</td>
</tr>
<tr>
<td>Bandpass filter</td>
<td>I-25, 51, 83, II-62, 63, 69</td>
<td>Bandpass filter</td>
</tr>
<tr>
<td>Blocked points</td>
<td>I-5, 42, 43, 48, 91, 92, 104:108, II-1, 11, 12, 75:78, B-3, 5, 11</td>
<td>Blocked points</td>
</tr>
<tr>
<td>Bottom level</td>
<td>I-27, 32, 34, II-10:12, 31, 43</td>
<td>Bottom level</td>
</tr>
<tr>
<td>Bottom levels</td>
<td>I-27, 32, 34, II-12, 31, 43</td>
<td>Bottom levels</td>
</tr>
<tr>
<td>CHI (Derived field)</td>
<td>I-57, 59, 70, 71, 77, 82, II-65, 66</td>
<td>CHI (Derived field)</td>
</tr>
<tr>
<td>CLBACK (ICP)</td>
<td>I-109, II-1</td>
<td>CLBACK (ICP)</td>
</tr>
<tr>
<td>CLCNTNT (ICP)</td>
<td>I-97, 99, 109, II-1</td>
<td>CLCNTNT (ICP)</td>
</tr>
<tr>
<td>CLCTRGE (ICP)</td>
<td>I-107, 109, II-1</td>
<td>CLCTRGE (ICP)</td>
</tr>
<tr>
<td>CLCTREQ (ICP)</td>
<td>I-107, 109, II-1</td>
<td>CLCTREQ (ICP)</td>
</tr>
<tr>
<td>CHIHIGH (ICP)</td>
<td>I-46, 107, 108, II-2</td>
<td>CHIHIGH (ICP)</td>
</tr>
<tr>
<td>CL LABEL (ICP)</td>
<td>I-46, 109, II-2</td>
<td>CL LABEL (ICP)</td>
</tr>
<tr>
<td>CLLOW (ICP)</td>
<td>I-46, 107, 109, II-2</td>
<td>CLLOW (ICP)</td>
</tr>
<tr>
<td>CLPNTVL (ICP)</td>
<td>I-106, 107, 109, II-2</td>
<td>CLPNTVL (ICP)</td>
</tr>
<tr>
<td>CLTABLE (ICP)</td>
<td>I-109, II-2, 3</td>
<td>CLTABLE (ICP)</td>
</tr>
<tr>
<td>CLSL (ICP)</td>
<td>I-46, 103, 104, 107, 109, II-3</td>
<td>CLSL (ICP)</td>
</tr>
<tr>
<td>CLVCED (ICP)</td>
<td>I-108, 109, II-3, 27, 28</td>
<td>CLVCED (ICP)</td>
</tr>
<tr>
<td>CLVLD (ICP)</td>
<td>I-108, 109, II-3, 27, 28</td>
<td>CLVLD (ICP)</td>
</tr>
<tr>
<td>COLOR (ICP)</td>
<td>I-109, II-3</td>
<td>COLOR (ICP)</td>
</tr>
<tr>
<td>CONH (Derived field)</td>
<td>I-57, 59</td>
<td>CONH (Derived field)</td>
</tr>
<tr>
<td>CONHILO (ICP)</td>
<td>I-97:101, 105:107, II-3</td>
<td>CONHILO (ICP)</td>
</tr>
<tr>
<td>CONLAB (ICP)</td>
<td>I-97, 107, 108, II-4</td>
<td>CONLAB (ICP)</td>
</tr>
<tr>
<td>.CONLABS (ICP)</td>
<td>I-97, 107, 108, II-4</td>
<td>.CONLABS (ICP)</td>
</tr>
<tr>
<td>.CONST (DERFLD operator)</td>
<td>II-10, 12:14, B-4, 6, 7, 18, 20</td>
<td>.CONST (DERFLD operator)</td>
</tr>
<tr>
<td>CONZERO (ICP)</td>
<td>I-97:102, 105:107, II-4</td>
<td>CONZERO (ICP)</td>
</tr>
<tr>
<td>CVFLDcn (ICP)</td>
<td>I-39, 84, II-4, 15, B-9, 12, 14</td>
<td>CVFLDcn (ICP)</td>
</tr>
<tr>
<td>Case comparison</td>
<td>I-85, (I-5, 15, 18, 19, 26, 36, 46, 47, 55, 84, 85, 87, 93, II-1, 15, 47, 49, 54, B-13)</td>
<td>Case comparison</td>
</tr>
<tr>
<td>Case differencing</td>
<td>I-56, 86</td>
<td>Case differencing</td>
</tr>
<tr>
<td>Case merging</td>
<td>I-55, 56, 87, B-2, 3, 11, 13</td>
<td>Case merging</td>
</tr>
<tr>
<td>Case merging</td>
<td>I-87, (I-5, 15, 26, 55, 56, 85, 86, II-13, B-2, 11, 13)</td>
<td>Case merging</td>
</tr>
<tr>
<td>Case ratios</td>
<td>II-10, 12</td>
<td>Case ratios</td>
</tr>
<tr>
<td>DASHLIN (ICP)</td>
<td>I-97:109, II-4, 73</td>
<td>DASHLIN (ICP)</td>
</tr>
<tr>
<td>DAYS (A)</td>
<td>I-5, (A-1:14, B-2, 4, 9:16)</td>
<td>DAYS (A)</td>
</tr>
<tr>
<td>DAYTPc (ICP)</td>
<td>I-6, 33, 45, II-6, 7, 15, 16, 31, 73, A-3, 10:14, B-10:15</td>
<td>DAYTPc (ICP)</td>
</tr>
<tr>
<td>DEBUG (ICP)</td>
<td>I-20, 21, II-7</td>
<td>DEBUG (ICP)</td>
</tr>
<tr>
<td>DEFLD (ICP)</td>
<td>I-54, 56, 57, 76, 84, 87, II-7, 8, B-9</td>
<td>DEFLD (ICP)</td>
</tr>
<tr>
<td>DELPRES (Derived field)</td>
<td>I-57, 60, II-53, B-5, 18</td>
<td>DELPRES (Derived field)</td>
</tr>
<tr>
<td>DELREL (ICP)</td>
<td>I-16, 18, 19, 33, 38, 40, II-8</td>
<td>DELREL (ICP)</td>
</tr>
<tr>
<td>DERFLD (ICP)</td>
<td>I-29, 45, 56, 76, 78, 84, 93, 108, II-8, 14, 66, A-10, 11, B-2:15, 18, 20</td>
<td>DERFLD (ICP)</td>
</tr>
<tr>
<td>DIFFL (ICP)</td>
<td>I-54, 56, 57, 60, 61, 85, 86, 102, II-15, 47</td>
<td>DIFFL (ICP)</td>
</tr>
<tr>
<td>DIV (Derived field)</td>
<td>I-57, 60, (A-2, B-6, 9, 10, 14, 15, 18:20)</td>
<td>DIV (Derived field)</td>
</tr>
<tr>
<td>:DIVIDE (DERFLD operator): II-11, 14, (B-6, 9, 10, 14, 15, 18:20)</td>
<td>:DIVIDE (DERFLD operator): II-11, 14,</td>
<td>:DIVIDE (DERFLD operator): II-11, 14,</td>
</tr>
<tr>
<td>:DKE (Derived field)</td>
<td>I-57, 60</td>
<td>:DKE (Derived field)</td>
</tr>
<tr>
<td>DLDAYS (ICP)</td>
<td>I-33, 40, II-6, 7, 15, 16</td>
<td>DLDAYS (ICP)</td>
</tr>
</tbody>
</table>

Index-1
<table>
<thead>
<tr>
<th>Index</th>
<th>November 1993</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLNPSX (Derived field): I-57,61,62,64,68,78, II-63</td>
<td></td>
</tr>
<tr>
<td>DLNPSY (Derived field): I-57,61,62,64,68,78, II-63</td>
<td></td>
</tr>
<tr>
<td>DP (Derived field): I-57,61 (B-10,18)</td>
<td></td>
</tr>
<tr>
<td>DPLTCA (ICP): I-95, II-16:20</td>
<td></td>
</tr>
<tr>
<td>DPLTCP (ICP): I-95, II-16,18,19</td>
<td></td>
</tr>
<tr>
<td>DPLTCT (ICP): I-95, II-17,18</td>
<td></td>
</tr>
<tr>
<td>DPLTFN (ICP): I-95,66, II-17:20</td>
<td></td>
</tr>
<tr>
<td>DPLTI (ICP): I-95, II-17,18</td>
<td></td>
</tr>
<tr>
<td>DPLTMF (ICP): I-10,11,95:97, II-16:19,36,43, B-10</td>
<td></td>
</tr>
<tr>
<td>DPLTRCP (ICP): I-95, II-18,19,20, B-10</td>
<td></td>
</tr>
<tr>
<td>DPLTRCPP (ICP): II-18,20</td>
<td></td>
</tr>
<tr>
<td>DRVRTYP (ICP): I-35,36,47, II-20,46,47,51,55,57</td>
<td></td>
</tr>
<tr>
<td>.DSTIMES (DERFLD operator): I-91, II-10,12,14, B-5,18</td>
<td></td>
</tr>
<tr>
<td>.DSWVSM (DERFLD operator): I-91, II-10,12, B-5</td>
<td></td>
</tr>
<tr>
<td>DVMAG (Derived field): I-57,61</td>
<td></td>
</tr>
<tr>
<td>Derived fields, Gradient: I-78, (I-76,77, II-63)</td>
<td></td>
</tr>
<tr>
<td>Derived fields, User-defined: I-76, (A-10,11, B-1,2,4,5,7,9,14)</td>
<td></td>
</tr>
<tr>
<td>Derived fields, Vector-pair: I-77, (I-56,76,77,82, II-65)</td>
<td></td>
</tr>
<tr>
<td>Derived fields: I-54, (I-6,23:26,28,31,34,48,81,106, II-7,8,20,21,63,66)</td>
<td></td>
</tr>
<tr>
<td>Disposition of plot file: I-95, (I-20,94, II-17,18,29,36,41:43,49)</td>
<td></td>
</tr>
<tr>
<td>EDOTDPDE (Derived field): I-57,62</td>
<td></td>
</tr>
<tr>
<td>.END (DERFLD operator): I-45,76, II-10,13,14,46,47,66, A-10,11, B-2,4,6:10,12,14,15,18:20</td>
<td></td>
</tr>
<tr>
<td>ENET (Derived field): I-57,62</td>
<td></td>
</tr>
<tr>
<td>ENETS (Derived field): I-57,62</td>
<td></td>
</tr>
<tr>
<td>ENETT (Derived field): I-57,62</td>
<td></td>
</tr>
<tr>
<td>ENSMBLc (ICP): I-33:35,43,44,102, II-5,20,58</td>
<td></td>
</tr>
<tr>
<td>:EQ (DERFLD operator): II-11,12,14, B-2,15</td>
<td></td>
</tr>
<tr>
<td>.EXP (DERFLD operator): II-10</td>
<td></td>
</tr>
<tr>
<td>Ensemble processing: I-34, (I-6,19,33,35, II-5,20)</td>
<td></td>
</tr>
<tr>
<td>Error handling: I-20, (I-5)</td>
<td></td>
</tr>
<tr>
<td>Extra bottom (surface) levels: I-32, (I-27,33,34,79,86,94, II-12)</td>
<td></td>
</tr>
<tr>
<td>FIELDcn (ICP): II-20</td>
<td></td>
</tr>
<tr>
<td>FLDSRcc (ICP): I-54,55, II-8,21</td>
<td></td>
</tr>
<tr>
<td>:GE (DERFLD operator): I-45, II-11,14, B-15</td>
<td></td>
</tr>
<tr>
<td>:GT (DERFLD operator): II-11, B-2,15</td>
<td></td>
</tr>
<tr>
<td>HEMIS (ICP): I-99, II-21, A-3,4, B-21</td>
<td></td>
</tr>
<tr>
<td>HPCDIV (ICP): I-97:100,109, II-1,2,21,22</td>
<td></td>
</tr>
<tr>
<td>HPCNWKI (ICP): I-97:99, II-23</td>
<td></td>
</tr>
<tr>
<td>HPCNWKR (ICP): I-97:99, II-23</td>
<td></td>
</tr>
<tr>
<td>HPDTCNT (ICP): I-97:99, II-23</td>
<td></td>
</tr>
<tr>
<td>HPMCINT (ICP): I-97,98, II-24</td>
<td></td>
</tr>
<tr>
<td>HPRADP (ICP): I-98:100, II-24</td>
<td></td>
</tr>
<tr>
<td>HPRASPR (ICP): I-97, II-24,26</td>
<td></td>
</tr>
<tr>
<td>HPRBND (ICP): I-98,99,106, II-24:26, B-10</td>
<td></td>
</tr>
</tbody>
</table>

Index-2
Index-3

November 1993

HPRSIZE (ICP): I-97, II-24,26
HPSCAL (ICP): I-97:99,100, II-26, B-20
HPVDIV (ICP): I-108,109, II-3,27, B-14
HTO (Derived field): I-57:59,63,74, II-50,70
HT1 (Derived field): I-57:59,63,74, II-15,21,49,50,64,70,75:77, B-6,7
HYBASGc (ICP): 1-33, II-29
HYBPRBc (ICP): 1-33, II-29
History Save Tapes, Output: I-41, (I-26,33,37,38,41,80, II-1,51,53,56,58, B-1)
History Save Tapes: 1-41, (1-42)
Horizontal Slice Save Tapes: I-43, (I-47)
Horizontal area averaging, Masked: I-92, (I-26,44,87,90,105, II-37,38,72, B-11)
ICP naming conventions: I-15, (I-5,21)
ICPECHO (ICP): I-94:96, II-29, B-10
INDEX (ICP): I-94:97, II-29,30,49,73
INTDP (ICP): I-78:80, II-30,50,51
INTDT (ICP): I-80, II-30,70
:INVINTB (DERFLD operator): II-11,13
:INVINTT (DERFLD operator): II-11,13, B-17:19
Input of History tapes: I-33, (I-33,38,41,78, II-53, A-1)
Introduction: I-2, (I-4)
Jobstepping: I-7,13
KE (Derived field): I-21,57,58,60,63,73,74
KTOOPV0 (Derived field): I-57,64
KTOOPV1 (Derived field): I-57,58,64,69
Kinetic Energy: I-57,58,60,63,70,71,73, A-9
:LITTIMES (DERFLD operator): II-11,14, B-7,18
LBTDP (ICP): I-78:80, II-30,50
LBTDT (ICP): I-80, II-31,70
:LE (DERFLD operator): II-11,12,14, B-4,6,7,10
LEAPY (ICP): I-33, II-6,31
.LEVELnn (DERFLD operator): I-29, II-10, B-9
:LEVMASK (DERFLD operator): II-11,12, B-4,6,7
LMLFPSL (ICP): I-100,101,107, II-31, A-7
LNPS (Derived field): I-57,61,62,64,65,68,78, II-63
LSD Save Tapes, Output: I-47, (I-26,47, II-65)
LSD Save Tapes: I-47, (I-35,37,38,43, II-20,46,47,51,54,55,57,71)
:LT (DERFLD operator): I-45, II-11
LXASPR (ICP): I-101, II-31,33
LXCDIV (ICP): I-101,109, II-1,2,31
LXCINT (ICP): I-28,29,101,102, II-32, A-8
LXLNRNG (ICP): I-101, II-32
LXLNSCL (ICP): I-60,81,101, II-32
LXPLOT (ICP): I-90,92,101,102, II-33
LXPTVAL (ICP): I-101,106,107, II-33
LXSCAL (ICP): I-101, II-33
LXSIZE (ICP): I-101, II-31,33
LYRSUBc (ICP): I-28,30,33,34,43,44, II-12,33,34,51
Limiting history tape input levels: I-34, (I-29,33, II-33,62)
List sorted data (LSD) processing: I-35, (I-19,22,33,35,47)
MASKc (ICP): I-87, II-34
MAGVDP (ICP): I-92, II-35, B-11,12
MAVPRN (ICP): I-92, II-35
MAVRNG (ICP): I-92,105,106, II-35,75, A-8,13, B-11,12
MAX (DERFLD operator): II-11
MBKFR (ICP): I-87,92,105,106, II-35,75
Memory (TOPIC) : I-30, (I-21,23:31, II-36)
MEMORY (ICP): I-21:30, II-36
MERAVG (ICP): I-92,101,102,106, II-32,33,35,36,75, A-8, B-11,12
MIN (DERFLD operator): II-11,14, A-10, B-7,8,9,10,14,20
MINUS (DERFLD operator): II-10
MINUS (DERFLD operator): II-11,14, A-10, B-7,8,9,10,14,20
MNFRMS (ICP): I-95,96, II-36
MQ (Derived field): I-57,58,65,69
MSDIRI (ICP): I-16,20, II-36
MSDIRic (ICP): I-16,20, II-36
MSDIRO (ICP): I-16,20, II-37
MSDIROc (ICP): I-16,20, II-37
MSKAP (ICP): I-92,93,104,105, II-37,38
MSKAPS (ICP): I-92,93,104,105, II-37,38
MSKAVPR (ICP): I-104,105, II-38
MSKAZS (ICP): I-92,93,104,105, II-37,38
MSKFlen (ICP): I-92,93,105, II-37:39,72
MSPPFXI (ICP): I-16,19, II-39,40,46,67
MSPPFXic (ICP): I-16,19, II-39,40,67
MSPPFXO (ICP): I-16,20,95,96, II-17,18,40,41,48,49,57
MSPPFXOc (ICP): I-16,20, II-40,41
MSRT0 (ICP): I-16,20,45, II-41,42,56,57
MSRT0c (ICP): I-16,20, II-41,42,56,57
MSTXT0 (ICP): I-16,20, II-42
MSTXT0c (ICP): I-16,20, II-42
MXASPR (ICP): I-100, II-43,45
MXCDIV (ICP): I-100,101,109, II-1,2,43
MXCINT (ICP): I-28,29,100,101, II-43, A-6
MXFRMS (ICP): I-95,96, II-43
MXLATRV (ICP): I-100, II-43,44
MXTNSCL (ICP): I-80,81,100, II-43
MXLTRNG (ICP): I-100,101, II-44
MXPTO (ICP): I-90,92,100,101,102, II-44
MXPTVal (ICP): I-100,101,106,107, II-44
MXSCAL (ICP): I-32,100,101, II-45
MXSIZE (ICP): I-100, II-43,45
MSS Use: I-16, (I-5,21,33,38,42, II-8,39:41,46,48,49)
Memory limits: I-27, (I-7,21,27,30,31,34,89, II-21)
Meridional averaging: I-92, (I-26,38,40,41,43,90,94,102,106, II-33,35,36, A-8, B-11)
NDYHSTc (ICP): I-41,42, II-45,57
NDYTSRc (ICP): I-40,41, II-45,60
:NE (DERFLD operator): II-11
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLCDP</td>
<td>(ICP): I-78-80, II-46,50</td>
<td></td>
</tr>
<tr>
<td>NLCDT</td>
<td>(ICP): I-80, II-46,70</td>
<td></td>
</tr>
<tr>
<td>NRADS</td>
<td>(Derived field): I-57,65</td>
<td></td>
</tr>
<tr>
<td>NSBDAYc</td>
<td>(ICP): I-35,36,47,48, II-46,47,55</td>
<td></td>
</tr>
<tr>
<td>NSBAPc</td>
<td>(ICP): I-35,36,47,48, II-47,55</td>
<td></td>
</tr>
<tr>
<td>NSDPRT</td>
<td>(ICP): I-65,66, II-47</td>
<td></td>
</tr>
<tr>
<td>NSVHSTc</td>
<td>(ICP): I-41, II-47,48</td>
<td></td>
</tr>
<tr>
<td>NSVTSCc</td>
<td>(ICP): I-40, II-48,49</td>
<td></td>
</tr>
<tr>
<td>NUMPLT</td>
<td>(ICP): I-94-96, II-29,49</td>
<td></td>
</tr>
<tr>
<td>OFTHSTc</td>
<td>(ICP): I-41,42, II-49,57</td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td>(DERFLD operator): II-11,12</td>
<td></td>
</tr>
<tr>
<td>ORIGFLD</td>
<td>(ICP): I-85,86, II-15,49</td>
<td></td>
</tr>
<tr>
<td>Order of Processor operations: I-22,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output of Horizontal Slice Save Tapes: I-43,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(I-26,37,38, II-55,56,58)</td>
<td></td>
</tr>
<tr>
<td>PINTXL</td>
<td>(ICP): I-78-80, II-30,46,50,51</td>
<td></td>
</tr>
<tr>
<td>PKHStc</td>
<td>(ICP): I-61,42, II-1-51,57</td>
<td></td>
</tr>
<tr>
<td>PKLS1c</td>
<td>(ICP): I-35,36,47,48, II-51</td>
<td></td>
</tr>
<tr>
<td>PLUS</td>
<td>(DERFLD operator): I-45, II-11,14, B-2,4,6,7,8,10,14,15,20</td>
<td></td>
</tr>
<tr>
<td>POWER</td>
<td>(DERFLD operator): II-11</td>
<td></td>
</tr>
<tr>
<td>PRECT</td>
<td>(Derived field): I-57,65</td>
<td></td>
</tr>
<tr>
<td>PRES</td>
<td>(Derived field): I-10,11,13,43,57,60,65,66,69,78,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>II-51,53,64,67,2-73,78,79, A-3,4,10,11,13,14, B-4:6-9,18:20</td>
<td></td>
</tr>
<tr>
<td>PRESp</td>
<td>(Derived field): I-57,66, B-18</td>
<td></td>
</tr>
<tr>
<td>PRESSLE</td>
<td>(ICP): I-10,11,13,43,78, II-51,64,67,73, A-3,4,10,11,13,14, B-4,9</td>
<td></td>
</tr>
<tr>
<td>PRFLDcn</td>
<td>(ICP): I-39,84, II-51,52, B-9</td>
<td></td>
</tr>
<tr>
<td>PRFNMc</td>
<td>(ICP): I-36, II-52</td>
<td></td>
</tr>
<tr>
<td>PRINTc</td>
<td>(ICP): I-36, II-52,53, B-12</td>
<td></td>
</tr>
<tr>
<td>PRLATc</td>
<td>(ICP): I-36, II-52</td>
<td></td>
</tr>
<tr>
<td>PRLIMRc</td>
<td>(ICP): I-21,36, II-53</td>
<td></td>
</tr>
<tr>
<td>PRLONc</td>
<td>(ICP): I-36, II-52,53</td>
<td></td>
</tr>
<tr>
<td>PRNTHDc</td>
<td>(ICP): I-33,36, II-53</td>
<td></td>
</tr>
<tr>
<td>PSI</td>
<td>(Derived field): I-46,57,66,70,71,77,82,103,104, II-65,66,71, B-17,18</td>
<td></td>
</tr>
<tr>
<td>PSL</td>
<td>(Derived field): I-57,66,67,100,101,107, II-31, A-7</td>
<td></td>
</tr>
<tr>
<td>PTOPc</td>
<td>(ICP): I-60, II-53, B-18</td>
<td></td>
</tr>
<tr>
<td>PWDSHTc</td>
<td>(ICP): I-41, II-53,54,57</td>
<td></td>
</tr>
<tr>
<td>PWDSTrc</td>
<td>(ICP): I-40, II-54,60, B-12</td>
<td></td>
</tr>
<tr>
<td>PWLS1c</td>
<td>(ICP): I-35,36,47, II-20,54</td>
<td></td>
</tr>
<tr>
<td>Plotting contours: I-107, (I-94,97:102,105,106, II-1:4,21,22,24,26,28,31:33,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>II-43:45,74,77, A-1,3,6,8,13</td>
<td></td>
</tr>
<tr>
<td>Plotting in color: I-109, (I-94,98,100,101,102,107, II-1,2,3,74)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plotting latitudinal cross sections: I-101, (I-26,80,92,94,107, II-31:33,43,81, A-8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plotting meridional avg. contours (long. vs. time): I-105,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(I-102, II-71,74,75, A-13)</td>
<td></td>
</tr>
<tr>
<td>Plotting meridional cross sections: I-100, (I-26,80,84,89,92,94,107, II-31:43,45,81, A-6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plotting rectangular horizontal projections: I-97, (I-97,107, II-25, A-1,2,5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Plotting time series: I-102, (I-23:26,93,94,107, II-3,4,59,71,73,76)
Plotting time vs. field value: I-103, (I-102, II-75, A-10)
Plotting time vs. masked area average: I-104, (I-44,93,102, II-71, A-12)
Plotting time vs. zonal average: I-103, (I-102, II-77, A-11)
Plotting zonal average contours (time vs. latitude): I-105, (I-102, II-76:78, A-14)
Printout of field values: I-36, (I-23:25, II-13,52,53, B-13)
Processing costs: I-28, (I-27,30)
Production of graphics: I-94, (I-92,100,101,109, II-2)
QSRC (Derived field): I-57,67
RADD (Derived field): I-58,67
RANDOM (DERFLD operator): II-10,12
RATFLDn (ICP): I-54,57,85,86,102, II-54
RELHUM (Derived field): I-58,67,68
RTLS1c (ICP): I-35,36,47,48,50,51, II-20,55,65
SAVFHSL (ICP): I-43,44, II-55
SAVHStc (ICP): I-41, II-37,45,47:49,53,56:58
SAVLS1c (ICP): I-35,36,47,48, II-20,54,57,71
SAVMHSL (ICP): I-43, II-56,57
SAVMHST (ICP): I-41,42, II-57,58
SAVNTSR (ICP): I-40, II-58
SAV TAVc (ICP): I-39, II-40:42,58,59, B-10
SAVTSPR (ICP): I-46,103, II-59
SAVTSPW (ICP): I-46,47,103, II-59,60
SAVTSTRc (ICP): I-40, II-45,48,49,54,58,60, B-12,17
SCRUB (ICP): I-16,17,19,33,38,40, II-60
SDFLDcn (ICP): I-39,84, II-8,60,61, B-9,13,14
SFCTCRT (ICP): I-45,87,88,92,93, II-61
SFCTTAP (ICP): I-45,87,88,92,93,104, II-61
SHIFTDN (DERFLD operator): II-10,12, B-4,7,8,20
SHIFTUP (DERFLD operator): II-10,12, B-7,8,20
SHSLZAV (ICP): I-43, II-56,61,62
SIGDOTH (Derived field): I-58,68,72
SIGDOTF (Derived field): I-58,68,71,72
SIGLEVe (ICP): II-34,62
SPCBPcn (ICP): I-81,83, II-62,63
SPCDFcn (ICP): I-81, II-63
SPCEFcn (ICP): I-31,55,64,81, II-63, B-20
SPCGRAD (ICP): I-54,56,78, II-63,64, B-20
SPCINTc (ICP): I-81,83, II-64, B-2,3
SPCMNKe (ICP): I-81,83, II-64, B-3
SPCTAVV (ICP): I-48,49,81,82, II-55,65
SPCV (ICP): I-54,56,77,78,82, II-65,66
SPGYINT (ICP): I-106, II-66
SQR (DERFLD operator): II-11,14, B-10,14
SURFLEV (ICP): I-28,30,32,33,43,44, II-66
Sample job deck: I-9,14, (I-7,8,96)
Save tapes: I-37, (I-29,33,35,80,84,85,90:92, II-6,66)
Spatial averaging: I-90, (I-36,91,92,94)
<table>
<thead>
<tr>
<th>Index</th>
<th>November 1993</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special contour levels for mixing ratio:</td>
<td>I-32, (I-27,107)</td>
</tr>
<tr>
<td>Special topics:</td>
<td>I-27</td>
</tr>
<tr>
<td>Specification of data to process:</td>
<td>I-33, (I-17,23:25, II-67)</td>
</tr>
<tr>
<td>Spectral bandpass filtering:</td>
<td>I-83, (I-25,81, II-62)</td>
</tr>
<tr>
<td>Spectral interpolation:</td>
<td>I-82, (I-25,78,81,86, II-64, B-1)</td>
</tr>
<tr>
<td>Spectral processing:</td>
<td>I-81, (I-25,26,28,55,59,76,78,82,83,95, II-62:64, B-3)</td>
</tr>
<tr>
<td>Surface Type Save Tapes, Input:</td>
<td>I-45</td>
</tr>
<tr>
<td>Surface Type Save Tapes, Output:</td>
<td>I-45</td>
</tr>
<tr>
<td>Surface type masking:</td>
<td>I-87, (I-23:25, II-71)</td>
</tr>
<tr>
<td>Spectral interpolation:</td>
<td>I-82, (I-25, 28, 81, 86, II-64, B-3)</td>
</tr>
<tr>
<td>Surface type Save Tapes:</td>
<td>I-44, (I-37,87,93, II-61,72, A-2)</td>
</tr>
<tr>
<td>Surface type Save Tapes:</td>
<td>I-45, (I-23:25, II-23,34)</td>
</tr>
<tr>
<td>TAPESc</td>
<td>(ICP): II-67</td>
</tr>
<tr>
<td>TCFLScn</td>
<td>(ICP): I-39,84,85, II-67</td>
</tr>
<tr>
<td>TCLD</td>
<td>(Derived field): I-58,68, A-12, B-15</td>
</tr>
<tr>
<td>.TCSL</td>
<td>(DERFLD operator): II-11,14, B-18</td>
</tr>
<tr>
<td>TEMPLEV</td>
<td>(ICP): I-80, II-67,68, B-19,20</td>
</tr>
<tr>
<td>TEST</td>
<td>(Derived field): I-58,68,69, II-71</td>
</tr>
<tr>
<td>TFWTSc</td>
<td>(ICP): I-88,89, II-68,69, B-17</td>
</tr>
<tr>
<td>THETA</td>
<td>(Derived field): I-58,69,80,81, II-78,79, B-19,20</td>
</tr>
<tr>
<td>TIMAVGc</td>
<td>(ICP): I-34,35,39,48,49,81,84,85,89,106, II-4,8,15,16,20,52,57,58,60, II-65,67,68,81,82, A-3,9, B-9,10,12,13,15,17</td>
</tr>
<tr>
<td>TIMES</td>
<td>(DERFLD operator): I-45, II-11,14,66, A-11, B-2,5,6,8,10,14,15,18,20</td>
</tr>
<tr>
<td>TIMFILc</td>
<td>(ICP): I-88,89, II-68,69, B-17</td>
</tr>
<tr>
<td>TINTMLT</td>
<td>(ICP): I-80, II-69</td>
</tr>
<tr>
<td>TINTXL</td>
<td>(ICP): I-80, II-30,31,46,69,70</td>
</tr>
<tr>
<td>TITLS1c</td>
<td>(ICP): I-35,36,47,52, II-71</td>
</tr>
<tr>
<td>TMQ</td>
<td>(Derived field): I-58,64,69</td>
</tr>
<tr>
<td>.TSINL</td>
<td>(DERFLD operator): II-11, B-20</td>
</tr>
<tr>
<td>.TSLPISP</td>
<td>(ICP): I-46,103,104, II-71</td>
</tr>
<tr>
<td>.TSLPSIZ</td>
<td>(ICP): I-46,103,104, II-71</td>
</tr>
<tr>
<td>TSMCASP</td>
<td>(ICP): I-46,105, II-71</td>
</tr>
<tr>
<td>TSMCSIZ</td>
<td>(ICP): I-46,105, II-71</td>
</tr>
<tr>
<td>TSPALcN</td>
<td>(ICP): I-46,104, II-37,38,71,73, A-12</td>
</tr>
<tr>
<td>TSPDYSc</td>
<td>(ICP): I-103, II-73, A-10:12</td>
</tr>
<tr>
<td>TSPFNPn</td>
<td>(ICP): I-46,103,104, II-73,74</td>
</tr>
<tr>
<td>TSPFPH</td>
<td>(ICP): I-46,103,104, II-73,74</td>
</tr>
<tr>
<td>TSPMCLR</td>
<td>(ICP): I-105,106, II-75</td>
</tr>
<tr>
<td>TSPMCCn</td>
<td>(ICP): I-46,105,106,109, II-1,2,74,75,77, A-13</td>
</tr>
<tr>
<td>TSPPLcn</td>
<td>(ICP): I-46,103, II-74:76, A-10</td>
</tr>
<tr>
<td>TSPYESCL</td>
<td>(ICP): I-103:105, II-76</td>
</tr>
<tr>
<td>TSP2CLR</td>
<td>(ICP): I-105, II-77</td>
</tr>
<tr>
<td>TSP2Cn</td>
<td>(ICP): I-46,105,109, II-1,2,76,77, A-14</td>
</tr>
<tr>
<td>TSP2Lcn</td>
<td>(ICP): I-46,103, II-77,78, A-11</td>
</tr>
<tr>
<td>.TSZCAS</td>
<td>(ICP): I-46,105, II-78</td>
</tr>
<tr>
<td>.TSZCSIZ</td>
<td>(ICP): I-46,105, II-78</td>
</tr>
<tr>
<td>TV</td>
<td>(Derived field): I-58,63,70,72,75,97:101,106,107,109, II-2,4,14,24,33,44,52,67,81, A-2, B-2,6</td>
</tr>
<tr>
<td>TYPEc</td>
<td>(ICP): I-4,21,33,34,38:40,45,54:57,80,87,92, II-5,53,78, A-2, B-12,17</td>
</tr>
<tr>
<td>Time Average Save Tapes, Input:</td>
<td>I-38, (I-38, A-2)</td>
</tr>
<tr>
<td>Time Average Save Tapes, Output:</td>
<td>I-39, (I-26,38)</td>
</tr>
</tbody>
</table>
| Time Average Save Tapes: | I-38, (I-27,29,37,44,84, II-58, B-16)
November 1993

Time Series Plot Save Tapes, Input: 1-46, (I-46,47)
Time Series Plot Save Tapes: I-46, (I-29,37,103, II-59)
Time Series Save Tapes, Output: I-40, (I-26,39)
Time Series Save Tapes: I-39, (I-29,37,38,89, II-54,60, B-11)
Time average statistics: I-84, (I-22,23,25,35,38:40,42,43,55,82,89,95,
II-4,20,51,60,67, B-8,14,16)
Time filtering: I-88 (I-22,24,25,28,40,42,43,47,56,84,95,102, II-68, B-17)
UD (Derived field): I-29,54,56:58,68,70,76,77,80,82,84,85,87,90,91,93,
I-105,108, II-1,8,10,38,39,50,65,66,70,72,76:79,
A-10:12, B-1,2,4,5,7,9,14,15
:UNBLOCK (DERFLD operator): II-11,12, B-4,5
UNDEFDF (ICP): I-54,56,57,76, II-79
UVSQ (Derived field): I-58,70,71, A-9
UVSQD (Derived field): I-58,70,71
UVSOR (Derived field): I-58,71
UZ (Derived field): I-58,71,77,82, II-65,66
Unblocking: B-5,6
Using the Processor: I-9, (I-5)
VADVGV1 (Derived field): I-58,71
VADVQVO (Derived field): I-58,71
VADVQV1 (Derived field): I-58,72
VADVTVV (Derived field): I-58,72
VADVUVV (Derived field): I-58,72
VARWV (Derived field): I-58,73
VAVCDSP (ICP): I-90,91, II-79
VAVRNG (ICP): I-90,91, II-79
VBKFR (ICP): 1-87,90,91, II-79
VD (Derived field): I-58,73,77,82,97:100,108,109,
II-3,11,12,14,27,28,65,66, A-4,5,11, B-14
VERAVG (ICP): I-90,91, II-79
VIKE (Derived field): I-58,73,74
VIPE (Derived field): I-58,73,74
VITE (Derived field): I-58,74
VMAG (Derived field): I-57,58,61,74
VOR (Derived field): I-58,66,70,71,73,74,77,82, II-65,66, B-19,20
.VSUM (DERFLD operator): I-91, II-11,14, B-5,6
VZ (Derived field): I-58,74,77,82, II-65,66
Vertical averaging: I-90, (I-26,38,40,41,43,90, II-22,79)
Vertical interpolation to potential temperature surfaces: I-80, (I-78,
II-30,31,46,69, B-5)
Vertical interpolation to pressure surfaces: I-78, (I-29,78,80,
II-30,46,50, A-3,10,11)
Vertical interpolation: I-78, (I-23:25,34,37,39,40,42,43,86, B-3)
Z (Derived field): I-58
Z1 (Derived field): I-58
Z2 (Derived field): I-58,75, II-14,15, A-3,10,13, B-9
.ZAV (DERFLD operator): II-11,12,14, A-11
.ZAVDEV (DERFLD operator): II-11,12,14, A-11
ZAVGDS (ICP): I-91,92, II-80, B-11,12

Index-8
ZAVGPRN (ICP): I-91, 92, II-80
ZAVRNG (ICP): I-91, 103, 104, 105, II-62, 77, 78, 80, A-14, B-11, 12
ZBKFR (ICP): I-87, 91, 92, 103, 104, 105, II-77, 78, 80
ZCVFLcn (ICP): I-39, 84, 85, 89, II-81, 82
ZONAVG (ICP): I-91, 100, 101, 104, 105, II-44, 62, 77:81, A-6, 7, B-11, 12, 18, 19
ZORD (ICP): I-100, 101, II-81
ZSBKFR (ICP): I-87, 89, II-81
ZSTFLcn (ICP): I-39, 84, 85, 89, II-82
Zonal averaging: I-91, (I-26, 38, 40, 41, 43, 90, 94, 101, 104, 105, II-44, 80, 81,
A-6, 7, 14, B-11)
Zonal eddy statistics: I-89, (I-23, 26, 84, II-81, 82, B-8)