CEDRIC

Custom **E**diting and **D**isplay of **R**educed
Information in **C**artesian space

National Center for Atmospheric Research *
Mesoscale and Microscale Meteorology Division
Boulder, Colorado 80307-3000

June 1998

*The National Center for Atmospheric Research is sponsored by the National Science Foundation.

CEDRIC

CEDRIC is a software program for the _Custom_ _Editing_ and _Display_ of _Reduced_ _Information_ in _Cartesian_ space. This manual describes the "Batch Version" written in Fortran 77 and C programming languages for mainframe and workstation computers having UNIX operating systems.

NOTICE

Published by the Mesoscale and Microscale Meteorology Division, P.O. Box 3000, Boulder, CO 80307-3000. The Mesoscale and Microscale Meteorology Division is a scientific division of the National Center for Atmospheric Research. The National Center for Atmospheric Research is operated by the University Corporation for Atmospheric Research and sponsored by the National Science Foundation. Any opinions, findings, conclusions, or recommendations expressed in this manual are those of its contributors, and do not necessarily reflect the views of the National Science Foundation.

This manual was prepared using TEX, the *Text EXecutive* processor developed by Honeywell Information Systems. TEX is a trademark of the American Mathematical Society.

Table of Contents

L. Jay Miller
(303) 497-8975
Email: ljmill@ncar.ucar.edu

Sherrie M. Fredrick
(303) 497-8973
Email: sherrie@ncar.ucar.edu

National Center for Atmospheric Research
Mesoscale and Microscale Meteorology Division
P.O. Box 3000
Boulder, Colorado 80307-3000

Leave this page blank

Leave this page blank

1. INTRODUCTION

The CEDRIC batch processor is designed primarily for the reduction and analysis of single and multiple Doppler radar volumes in either a Cartesian or a coplane coordinate system. CEDRIC provides a wide variety of options for calculating physical quantities of interest, for editing the data, for filtering, as well as options for displaying the data graphically. Users have flexibility in manipulating fields by building function stacks from algebraic primitives. Also users can obtain color area-filled plots and overlay colored vectors or other contoured fields. The plot files can be easily examined on workstations too, using utilities that are part of the NCAR Graphics Package (GKS) from the Scientific Computing Division (SCD).

CEDRIC requires input data files that are structured in a very specific way. This format is described in some detail in Appendix D of this manual and in the SPRINT documentation. The data files can be the output from other programs such as interpolators like SPRINT, or the output files created by previous CEDRIC runs. The most important requirement for data input to CEDRIC is that it be regularly-gridded data (no staggered grids), and in a format understood by CEDRIC. We provide an input-output software package for reading-writing CEDRIC format for users who wish to interface CEDRIC with their own analysis packages or model data outputs.

Current memory sizing allows data sets that meet the following conditions: a) number of grid points along any one axis msut be less than 512, b) number of grid points in a horizontal plane (NX by NY) must not exceed 65536, c) only 12 z-levels are allowed if NX*NY=65536, and d) number of fields allowed is 25. The total number of grid points NX*NY*NZ cannot exceed 786432. More than 12 z-levels are allowed if fewer than 65536 horizontal points are used, but the absolute maximum is 128. The absolute memory limit for the product of grid points and number of fields, NX*NY*NZ*NF, is 8000000.

CEDRIC computer runs consist of three basic parts. The first part is the input stage, where data is read by CEDRIC into an edit volume. The second part is the editing and analysis stage that includes decimation, filling, filtering, as well as multiple Doppler radar wind synthesis and algebraic manipulation of fields to produce other physical quantities of interest. Finally, some kind of output is generated that can be graphical, statistical and numerical output. Additionally, new edit volumes are available for output and later use as input to other CEDRIC runs. Although CEDRIC was designed primarily for multiple-Doppler radar wind analysis, it is very useful for other kinds of gridded data such as from aircraft (ACANAL), mesonetwork (GEMINI) and numerical models.

CEDRIC was written originally by Carl Mohr under the supervision and guidance of L. Jay Miller. Bill Anderson converted that software to the Cray-YMP and has added considerable enhancements such as color area-fill graphics and a complete coplane analysis. Bill is currently responsible for the continued programming aspects of the project, and L. Jay Miller is responsible for the overall direction and the scientific aspects. We request that you acknowledge NCAR/MMM as a source of this software package when it is used or referenced in any resulting research, publications, or subdistributions.

The following pages contain a description of the commands that have been implemented in the batch version of CEDRIC, running either on the Cray-YMP or one of several Unix workstations (DEC, SUN, IBM and SGI). The CEDRIC commands are identical in these two environments; however, the Unix shell scripts are not. Though this document concentrates on the Cray version since the shell scripts for workstations may be system dependent, most of what is documented here applies equally well to the Unix workstation versions. As with any large software package there are bound to be errors that have escaped our scrutiny. Please report any annoying or inconsistent program behavior so that appropriate corrections can be made.

Included in Appendix F is an in-depth review of multiple-Doppler wind synthesis. The basic steps involved in this analysis are also included here as introduction to this software package. The user should also refer to Appendix F for further details. The equations referenced here are within that Appendix.

The following steps are taken for multiple-Doppler radar wind synthesis:

1. Interpolate the radar measurements using the SPRINT software package. If measurements were originally taken in the coplane coordinate system, interpolation within these coplanes is selected. If measurements were taken in the normal radar spherical coordinate system, interpolation to Cartesian coordinates should be done. There is little advantage in interpolating to coplanes if the data were not taken in this coordinate system. The SPRINT package has no provision for such interpolation; however, the REORDER package of ATD/RDP does.
2. Select the appropriate coordinate system synthesis, integration and interpolation path in CEDRIC. This is done automatically if the appropriate housekeeping words are correctly set. Otherwise, the user must invoke the COORD command to force the correct choice of mathematical formulation.
3. Specify the necessary parameters in the SYNTHES command in CEDRIC. The Cartesian, orthogonal particle-motion solutions (U,V,W) are derived from three (or more) equations, relating non-orthogonal radial velocities to the (U,V,W) components. The Cartesian synthesis also allows for an over-determined or least-squares solution for (U,V) in terms of measured radial velocities, associated radar geometries, and an unknown vertical velocity contribution. For coplane synthesis at coplane coordinates, only two radars are allowed. Solutions are obtained for orthogonal, coplane components that are parallel and perpendicular to the two-radar baseline.
4. Invoke the CONVERGE command to compute horizontal or coplane convergence, the negative of divergence of the horizontal winds from the solutions in step 3 above.
5. Specify the necessary parameters in the INTEGR command for Cartesian or coplane integration integration of the mass continuity equation. When the W-contaminated (U,V) solutions are obtained, one way to solve the mass continuity equation is an iterative scheme implemented in the MASS2 command.

2. REFERENCES

These references are mainly for the CEDRIC software package. Other references on the details of multiple-Doppler radar wind synthesis are included in Appendix F.

Miller, L.J. and R.G. Strauch, 1974: A Dual Doppler Radar Method for the Determination of Wind Velocities within Precipitating Weather Systems. *Remote Sensing of Environment*, **3**, 219-235.

Miller, L.J., C.G. Mohr and A.J. Weinheimer, 1986: The simple rectification to Cartesian space of folded radial velocities from Doppler radar sampling. *J. Atmos. and Ocean. Tech.*, **3**, 162-174.

Mohr, C.G., L.J. Miller and R.L. Vaughan, 1981: An interactive software package for the rectification of radar data to three- dimensional Cartesian coordinates. *Pre- prints 20th Conf. on Radar Meteorology, Boston, Amer. Meteor. Soc.*, 690-695.

—— L.J. Miller, R.L. Vaughan and H.W. Frank, 1986: The merger of mesoscale datasets into a common Cartesian format for effiecient and systematic analyses. *J. Atmos. and Ocean. Tech.*, **3**, 144-161.

Rew, R.K. and G.P. Davis, 1990: NetCDF: An Interface for Scientific Data Access. *Computer Graphics and Applications, IEEE*, 76-82.

3. DESCRIPTION OF THE COMMAND SYNTAX

CEDRIC commands consist of 80 character card image formats beginning with a keyword starting in column 1. Each card image is divided into 10, 8-character wide fields referred to in the documentation as parameter positions P1 thru P10. The keyword always occupies field 1. Fields 2-10 contain required parameters that are interpreted by the program as either floating-point (F8.0) or alphanumeric (A8) entities. These are designated as (F) and (A) respectively. (F) parameters *must* be right justified within the field if they do not have a decimal point. (A) parameters *must* always be left justified. There also exists a hybrid entity (H) that is first examined as an (A) parameter and if it does not match a predetermined mnemonic, it is interpreted as an (F) parameter. Default values are used whenever a parameter field is left blank or if the information specified is erroneous. If a default value is not specified in the documentation, "blanks" will be supplied for an (A) field and 0.0 will be supplied for an (F) field.

Documentation for each command is as follows:

COMMAND—A brief description of the command, including how it might be used.

Command structure:

```
COMMAND P2      P3      P4                P6                       P9      P10
```

Parameter	Type	Name	A brief description of the parameter,
P2	(A)	AVALUE	A brief description of this parameter.
P3	(F)	FVALUE	A brief description of this parameter.
P4	(H)	HVALUE	A brief description of this parameter.
P5		UNUSED	++++++++++++++++++++
P6	(A)	AVALUE	A brief description of this parameter.
P7-8		UNUSED	++++++++++++++++++++
P9	(F)	FVALUE	A brief description of this parameter.
P10	(A)	IWINDO	Windowing specification for this command:

Commands are processed in the order in which they appear and may be repeated as often as desired. Certain commands (READVOL and SYNTHES) that generate new edit files will reset the spatial and field windowing to protect the user from inadvertently editing and displaying information outside the boundaries of the dataset. Some commands [FIELDSET, GRAPHICS (when specifying contours), MASS2, and TRANSFER] may require additional information that cannot be contained on a single card image. These are referred to as multiple-card commands and whenever any additional fields are necessary, EXTRA card images will be processed for the required information until a "blank" field is encountered. Termination by an END card image is not required. EXTRA is not a mandatory keyword, but instead refers to the requirement for additional card images whenever it appears in the documentation.

Other commands (CODED, DIGITAL, FUNCTION, HISTO, GRAPHICS, REMAP, STPLOT, SURFACE, SYNTHES, THREED and WTRANS) initiate the formation of a *stack*, consisting of multiple card images that contain additional information required by the command. These stacks *must* be terminated by an END command. END is a required mnemonic and must appear in the command keyword field P1. The COMMENT command also initiates a stack. The COMMENT stack must be terminated by a COMEND command. Any card image containing an asterisk "*" in column one (1) will also be treated as a comment no matter where it appears in the deck.

Facilities also exist for establishing a set of commands as a single entity using the DEFINE command, substituting user-defined variables with the VALUE command and executing the command set as often as needed (EXPAND). The DEFINE command initiates the formation of a stack consisting of valid CEDRIC commands. This stack *must* be terminated by a DEFEND command in order to differentiate it from other stacks. DEFEND is a required mnemonic and must appear in the command keyword field P1.

The only commands that invoke fully three-dimensional manipulation of data are FILTER (options "L3D" for Leise 3-D filtering and "LT3" for least-squares 3-D filtering), PATCHER (options "FILLCON3" and "FILLALL"), the SURFACE and THREED commands for plots of isosurfaces, and REMAP since data will be interpolated to a new three-dimensional grid. The way the program manipulates the data for these commands is controlled internally. All other commands are inherently two-dimensional, where operations are performed on successive XY-, XZ- or YZ-planes of data while moving in the Z, Y, or X direction.

When the direction for moving through the data can be specified by the user, it is set with the parameter "IFIXAX", usually P3 or P9. The only commands where the direction is setable are CROSS, FILTER, FUNCTION, GRAPHICS, HISTO, STATS, STPLOT, and UNFOLD. All other commands are conveniently done in two-dimensional XY-planes. But, since the result is not affected by the direction through the data, the user needs no control over direction (fixed axis). For graphical output and statistical output, direction controls the display plane; for example, Z-direction allows displays of XY-planes. The direction to pass through the data is implicit in some commands like INTEGR to integrate the mass continuity equation for the vertical component of air motion.

4. SUMMARY OF CEDRIC COMMANDS

Command	**Stack**	**Description of its functionality**
BADVAL	No	Specifies the floating point value for missing data flag
BCKGRND	No	Allows the user to force the background color for plots.
CHANGE	No	Alters the value at a specific (X,Y,Z) location
CODED	Yes	Generates an alphanumeric display of data fields
COMMENT	Yes	Transmits user-supplied comments to a print file
COMEND	No	Terminates a set of user-supplied comments
CONVERGE	No	Calculates horizontal convergence of (U,V) winds
COORD	No	Specifies coordinate system that data is in
CREATE	No	Defines a user-specified Cartesian coordinate system
CROSS	No	Computes the cross correlation between a pair of fields
DEFINE	Yes	Initiates the formation of a command block
DEFEND	No	Terminates the definition of a command block
DIGITAL	Yes	Generates a digital display of data fields
DELETE	No	Deletes a field from the current edit file
END	No	Terminates a stack operation
EXPAND	No	Executes a command block
FIELDSET	No	Partitions the current set of fields into 2 groups
FILTER	No	Performs spatial filtering (2- or 3D) of data fields
FIXIDS	No	Alters information in the 510-word volume header
FLTERTH	No	Enables/disables flat earth mode
FUNCTION	Yes	Algebraic manipulation (2D) of data fields
GRAPHICS	Yes	Generates 2-D plotted displays of data fields
HISTO	Yes	Generates histogram displays of data fields
INTEGR	No	Performs vertical integration of a data field
LAPLACE	No	Least-squares solution of a 2-D differential equation

Command	Stack	Description of its functionality
MASS2	No	Integrates mass continuity equation for 2-radar synthesis
PATCHER	No	Performs 2- or 3-D filling and 2-D decimation of a data field
QUIT	No	Terminates the CEDRIC run
READAIR	Yes	Transfers an ASCII file of aircraft track data into CEDRIC for use later in GRAPHICS
READSTA	No	Transfers an ASCII file of station locations for use in GRAPHICS
READVOL	No	Transfers a Cartesian radar volume into CEDRIC
REGRESS	No	Performs linear regression of one field versus another
RELAXUV	No	Modify (U,V) components using variationally adjusted W
REMAP	Yes	Remaps all existing fields to a new coordinate system
RENAME	No	Renames a field in the current edit file
SAMPLER	No	Derives radial velocity from a set of (U,V,W) winds
SHIFTER	No	Advects data fields in horizontal direction
STATS	No	Generates a statistical display for a set of fields
STPLOT	Yes	Generates plotted profiles of selected statistics
SURFACE	Yes	Generates 3-D perspective displays of Z-plane data fields
SYNTHES	Yes	Synthesizes radial velocity estimates from multiple radars
THREED	Yes	Generates 3-D isosurface displays of data fields
TRANSFER	No	Transfers fields from another CEDRIC volume to the edit file
UNFOLD	No	Performs de-aliasing of radial velocity measurements
VALUE	No	Permits the parameterization of CEDRIC commands by the user
WINDOW	No	Defines a user-specified spatial window for editing and display
WRITVOL	No	Transfers the current edit file to an external storage device
WTRANS	Yes	Windowed field transfer with spatial translation of input

DETAILS of CEDRIC COMMANDS

Custom **E**diting and **D**isplay of **R**educed
Information in **C**artesian space

BATCH PROCESSOR for UNIX-BASED COMPUTERS

Leave this page blank

Leave this page blank

BADVAL—Specifies the floating point value to which missing data is set. This command is useful for generating plausible contour lines between regions with data and regions without data. For example, any missing data surrounding good radar reflectivities could be set to a value just below the lowest true data value in order to close contours. *Note: This is a single card image command.*

Command structure:
`BADVAL  BAD`

P2	(F)	BAD	Missing data points will be set to this value. (DEFAULT = `-1000.0`)
P3-10		UNUSED	++++++++++++++++++++

BCKGRND—Sets the background color for plots. By default, the background color for grey scale plots is white and the background for all other plots is black. *Note: This is a single card image command.*

Command structure:
`BCKGRND COLOR`

P2	(A)	COLOR	Color to which to set background. Only `WHITE` and `BLACK` can be specified. `WHITE` indicates that the background color for all plots should be white. `BLACK` indicates that the background color for all plots except grey scale plots should be black.

CHANGE—Sets a specified field at a designated (X,Y,Z) location to a user-supplied value. This command is useful for setting certain data points to desired values or for eliminating data glitches. It is not recommended for extensive editing. *Note: This is a single card image command.*

Command structure:

```
CHANGE  NAMFLD  ISPCNX  XLOC    YLOC    ZLOC    VALUE
```

P2	(A)	NAMFLD	Name of the field to be changed.
P3	(A)	ISPCNX	Context within which the (X,Y,Z) location is defined: `INDEX` coordinates are specified in index space; `DISTANCE` coordinates are specified in distance space; (DEFAULT = `INDEX`)
P4	(F)	XLOC	X coordinate of the data point to be altered.
P5	(F)	YLOC	Y coordinate of the data point to be altered.
P6	(F)	ZLOC	Z coordinate of the data point to be altered.
P7	(H)	VALUE	Location (XLOC,YLOC,ZLOC) in NAMFLD is set to this value. If VALUE = `BAD`, NAMFLD at (XLOC,YLOC,ZLOC) will be set to the missing data flag (see the **BADVAL** command). (DEFAULT = `0.0`)
P8-10		UNUSED	++++++++++++++++++++

CODED—Generates a coded display (one alphanumeric symbol per data value) for a set of designated fields. This is a stack operation where additional card images allow the user to specify the fields to be examined along with associated display parameterization. The stack *must* be terminated by an **END** command. If a field is referred to more than once, the last specification will be used to display that field. Fields will be displayed according to their order within the CEDRIC program and not according to their order in the stack.

Command structure:

```
CODED   IDEST   IFIXAX  SKIP                                              IWINDO
        NMSPEC  SCLFAC  ZREF    WIDTH   NLFT    NRGT
END
```

P2	(A)	IDEST	Destination of the display: `PRINT` display will be sent to the printer file; `MICRO` display will be sent to microfiche/microfilm; `BOTH` display will be sent to both printer and film. The only available option is `PRINT`; the others have been documented here for historical reasons. (DEFAULT = `PRINT`)
P3	(A)	IFIXAX	Axis to be fixed for displays: `Z` z-axis fixed, display xy-planes; `Y` y-axis fixed, display xz-planes; `X` x-axis fixed, display yz-planes. (DEFAULT = `Z`)
P4	(F)	SKIP	Increment between display levels. A display will be generated every SKIP levels. (DEFAULT = `1.0`)
P5-9		UNUSED	++++++++++++++++++++
P10	(A)	IWINDO	Windowing specification for the display: `WINDOW` display will be confined to the current user-set window; `FULL` display will be generated over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

CODED –STACK of ADDITIONAL CARD IMAGES–

P2	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` only the primary field group will be displayed; `SEC` only the secondary field group will be displayed; `ALL` every field will be displayed. See the **FIELDSET** command for instructions on how to set primary and secondary field groups. (DEFAULT = `PRI`)
P3	(F)	SCLFAC	Scaling factor. Field values will be multiplied by this number before they are encoded. (DEFAULT = `1.0`)

P4	(F)	ZREF	Reference value for display. All field values less than ZREF will be encoded as lower-case letters; all field values greater than ZREF will be represented as upper-case. Values beyond the boundaries set by the user will be represented by either plus (+) or minus (–), depending on their sign. Missing data values will be represented by a period (.) (DEFAULT = `0.0`)
P5	(F)	WIDTH	Width for the individual bins. All data points within each of these intervals will be assigned the same **CODED** symbol. (DEFAULT = `5.0`)
P6	(F)	NLFT	Number of bins to the left of reference value (0-26) (DEFAULT = `0.0`)
P7	(F)	NRGT	Number of bins to the right of reference value (0-26) (DEFAULT = `0.0`)
P8-10		UNUSED	++++++++++++++++++++

END This **END** command terminates the **CODED** command stack.

COMMENT—Transmits user-supplied comments to an output file. This is a stack operation where additional card images contain user-supplied comments, one card image per line of text. The stack *must* be terminated by a **COMEND** command. Any CEDRIC commands appearing between the **COMMENT** and **COMEND** commands will be treated as comments and will not be executed. This command is useful for including comments about the data flow and commands, or to comment out large groups of commands that are not to be executed such as when the setup is being debugged. The latter would be done by simply surrounding the desired group with a **COMMENT-COMEND** combination.

Command structure:

```
COMMENT IDEST
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
COMEND
```

P2	(A)	IDEST	Destination of the user-supplied comments: `PRINT` comments will be sent to the printer file; `MICRO` comments will be sent to microfiche/microfilm; `BOTH` comments will be sent to both printer and film. The only available option is `PRINT`; the others have been documented here for historical reasons. (DEFAULT = `PRINT`)

COMMENT	–STACK of ADDITIONAL CARD IMAGES–

P1-10	(A)	TEXT	A line of user-supplied text; include as many card images as desired. CEDRIC does allow comment text in the P1 location; however, it has not been included in the above *Command structure* for readability.

COMEND This **COMEND** command terminates the **COMMENT** command stack.

COMEND—Terminates a stack of comments. It is used only in conjunction with the **COMMENT** command, and *must* appear following the last comment card image in the **COMMENT** stack. *Note: This command is itself a single card image command, and it requires no parameters.*

Command structure:

```
COMEND
```

CONVERGE—Calculates the two-dimensional convergence of the horizontal Cartesian or coplane two-dimensional wind components. The convergence is calculated using either 3- or 5-point centered differencing in either Cartesian space or in coplane space, depending on the coordinate system of the data as determined from the volume header or from the **COORD** command. See **Appendix F** for further details on the Cartesian and coplane forms of two-dimensional convergence and their use in the mass continuity equation. *Note: This is a single card image command.*

Command structure:
`CONVERGENAMOUF  NAMU    NAMV    NDER`

P2	(A)	NAMOUF	Name of the output two-dimensional convergence field.
P3	(A)	NAMU	Name of the input U-component of the wind field.
P4	(A)	NAMV	Name of the input V-component of the wind field.
P5	(F)	NDER	Number of points along each axis to use when computing the convergence estimate. The only acceptable values are `3.0` and `5.0` (DEFAULT = `3.0`)
P6-10		UNUSED	++++++++++++++++++++

COORD—Forces CEDRIC to interpret the data of all subsequent volumes as being in a particular coordinate system. The coordinate system chosen will affect the way that the synthesis **SYNTHES**, convergence **CONVERGE**, and integration **INTEGR** calculations are carried out. See **Appendix F** for the details of calculations performed in Cartesian space and coplane space. Use of this command also will set ID words 16 and 17 (coordinate system words) to either `CRT` for Cartesian or `COPL` for coplane for any volume that CEDRIC *currently* has in memory. If this command is not used, CEDRIC will obtain coordinate system information from the volume header (ID words 16 and 17). This command stays in effect until either the **COORD** command is called again or until the end of the CEDRIC script. *Note: This is a single card image command.*

Command structure:
`COORD   ICORD`

P2	(A)	ICORD	Coordinate system options: `CART` assume the data is in a three-dimensional Cartesian coordinate system regardless of the information in the volume header; `COPLANE` assume the data is in a coplane coordinate system regardless of the information in the volume header. There is no DEFAULT value. If this command is included, the user must specify the coordinate system.
P3-10		UNUSED	++++++++++++++++++++

CREATE—Defines or deactivates a user-supplied Cartesian coordinate system into which is mapped all subsequent **INPUT** volumes and **SYNTHESIZED** output volumes. This command is particularly useful when synthesizing multiple Doppler radar CEDRIC volumes that have not been interpolated to identical Cartesian grid systems. Any Cartesian coordinate systems to be remapped *must* have some grid locations in common with the **CREATEd** coordinate system. *No other restrictions are imposed.* The user-defined (X,Y,Z) grid is specified in kilometers (km) relative to some predetermined origin in (X,Y) horizontal space and km relative to Mean Sea Level (MSL) along the Z–axis. *Note: This is a single card image command.*

Command structure:

```
CREATE  X1      X2      XD      Y1      Y2      YD      Z1      Z2      ZD
```

P2	(H)	X1	If `X1` = `OFF`, mapping of INPUT and SYNTHESIS volumes into **CREATEd** coordinates will be deactivated. Otherwise, X1 is interpreted as the beginning X-coordinate of the user-defined (X,Y,Z) grid.
P3	(F)	X2	Ending X-coordinate.
P4	(F)	XD	Grid spacing along the X-axis.
P5	(F)	Y1	Beginning Y-coordinate of the user-defined (X,Y,Z) grid.
P6	(F)	Y2	Ending Y-coordinate.
P7	(F)	YD	Grid spacing along the Y-axis.
P8	(F)	Z1	Beginning Z-coordinate of the user-defined (X,Y,Z) grid.
P9	(F)	Z2	Ending Z-coordinate.
P10	(F)	ZD	Grid spacing along the Z-axis.

CROSS—Computes the two-dimensional cross correlation function between two-dimensional planes of a pair of fields that are lagged with respect to one another to a maximum specified distance along each axis in two-dimensions. Any axis may be held fixed, and the cross correlation function will be computed only in those regions within the designated geometric window. *Note: This is a single card image command.*

Command structure:

```
CROSS   NAMOUF  NAMREF  NAMLAG  LAGI    LAGJ    IZERO   JZERO   IFIXAX  IWINDO
```

P2	(A)	NAMOUF	Name of the cross correlation output field.
P3	(A)	NAMREF	Name of the reference input field.
P4	(A)	NAMLAG	Name of the input field to be lagged with respect to NAMREF. (DEFAULT = If either NAMREF or NAMLAG are left blank the input field that was specified will be lagged with respect to itself, the auto-correlation function.)
P5	(F)	LAGI	Number of grid locations to lag in a positive and negative direction along the current I-direction. LAGI must not exceed the default value. (DEFAULT = Number of I-locations within the window divided by 4)
P6	(F)	LAGJ	Number of grid locations to lag in a positive and negative direction along the current J-direction. LAGJ must not exceed the default value. (DEFAULT = Number of J-locations within the window divided by 4)
P7	(F)	IZERO	Specifies the I-index in the output field where the zero-lag correlation coefficient is to be located. (DEFAULT = Results will be stored, beginning at the left boundary of the output array.)
P8	(F)	JZERO	Specifies the J-index in the output field where the zero-lag correlation coefficient is to be located. (DEFAULT = Results will be stored beginning at the lower boundary of the output array.)
P9	(A)	IFIXAX	Axis to be fixed for two-dimensional cross correlation: `Z` z-axis fixed, correlation is done in xy-planes; `Y` y-axis fixed, correlation is done in xz-planes; `X` x-axis fixed, correlation is done in yz-planes. (DEFAULT = `Z`)
P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

DEFINE—Initiates the formation of a command block. A command block may consist of any valid sequence of CEDRIC commands except another **DEFINE**. This is a unique stack operation and *must* be terminated by the **DEFEND** command. This sequence of commands is saved for later expansion and execution. None of the commands are actually executed until the block is invoked using the **EXPAND** command. **EXPAND** commands may be contained within **DEFINE** blocks provided that the nesting level does not exceed five (5). A maximum of 25 **DEFINE** blocks may be created during the CEDRIC run, and the total number of card images contained in all the existing **DEFINE** blocks may not exceed 10000. A **DEFINE** block may not be redefined once it has been established. An example using CEDRIC commands thus far documented is shown in the command structure.

Command structure:

```
DEFINE  NAMBLK
COMMENT IDEST
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
COMEND
CREATE  X1      X2      XD      Y1      Y2      YD      Z1      Z2      ZD
BADVAL  BAD
CHANGE  NAMFLD  ISPCNX  XLOC    YLOC    ZLOC    VALUE
CODED   IDEST   IFIXAX  SKIP                                            IWINDO
        NMSPEC  SCLFAC  ZREF    WIDTH   NLFT    NRGT
END
COORD   ICORD
CONVERGENAMOUF  NAMU    NAMV    NDER
CROSS   NAMOUF  NAMREF  NAMLAG  LAGI    LAGJ    IZERO   JZERO   IFIXAX  IWINDO
DEFEND
```

P2	(A)	NAMBLK	The name of the **DEFINE** block to be associated with the set of CEDRIC commands contained in the stack. (DEFAULT = The user is permitted to have one unnamed **DEFINE** block.)

DEFINE	–STACK of ADDITIONAL CARD IMAGES–

P1-10	(H)	CEDRIC commands as they would appear if they were actually being executed.

DEFEND This **DEFEND** command terminates the **DEFINE** command stack.

DEFEND—Terminates the DEFINE block. It is used only in conjunction with the **DEFINE** command and must always appear following the last entry in the stack. A **DEFEND** command must always be present even if the **DEFINE** block contains no entries. *Note: This command is itself a single card image command, and it requires no parameters.*

Command structure:

```
DEFEND
```

DIGITAL—Generates a digital display (fixed number of digits per value) for a set of designated fields. This is a stack operation and *must* be terminated by an **END** command. The stack consists of additional card images listing the fields to be displayed and the specifications of the displays. Only one specification per field is permitted in the stack. If a field is referred to more than once, the last specification will be used to display that field. Fields will be displayed according to their order within the CEDRIC program and not according to their order in the stack.

Command structure:

```
DIGITAL IDEST   IFIXAX  SKIP                                              IWINDO
        NMSPEC  SCLFAC  NDIG
END
```

P2	(A)	IDEST	Destination of the display: The only available option is `PRINT`.
P3	(A)	IFIXAX	Axis to be fixed for displays: `Z` z-axis fixed, display xy-planes; `Y` y-axis fixed, display xz-planes; `X` x-axis fixed, display yz-planes. (DEFAULT = `Z`)
P4	(F)	SKIP	Increment between display levels. A display will be generated every SKIP levels. (DEFAULT = `1.0`)
P5-9		UNUSED	++++++++++++++++++++
P10	(A)	IWINDO	Windowing specification for the display: `WINDOW` display will be confined to the current user-set window; `FULL` display will be generated over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

DIGITAL –STACK of ADDITIONAL CARD IMAGES–

P2	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` only the primary field group will be displayed; `SEC` only the secondary field group will be displayed; `ALL` every field will be displayed. See the **FIELDSET** command for instructions on how to set primary and secondary field groups. (DEFAULT = `PRI`)
P3	(F)	SCLFAC	Scaling factor. Field values will be multiplied by this number before they are displayed. (DEFAULT = `1.0`)
P4	(F)	NDIG	Maximum number of digits allocated for each value. Values which cannot be displayed using the number of digits allocated will be represented as either plus (+) or minus (–) depending upon their sign. Missing data values will be represented by a period (.) (DEFAULT = `3.0`)
P5-10		UNUSED	++++++++++++++++++++

END This **END** command terminates the **DIGITAL** command stack.

DELETE—Deletes fields from the list of those present in the current edit file. If a non-existent field is designated for deletion a warning message will be provided. *Note: This is a single card image command.*

Command structure:

```
DELETE  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL
```

P2-10	(A)	NAMDEL	Names of the fields to be deleted.

END—Terminates a stack operation. It is used only in conjunction with stack definitions and *must* always appear following the last entry in the stack. An **END** command must always be present even if the stack contains one or no entries. **END** *cannot be used* to terminate CEDRIC processing. **QUIT** is the appropriate command for program termination. *Note: This command is itself a single card image command, and it requires no parameters.*

Command structure:

```
END
```

EXPAND—Executes the set of commands associated with the named **DEFINE** block. All CEDRIC commands within each block that is expanded will be inserted into the command stream where the **EXPAND** appears. Any **DEFINE** block may be **EXPANDed** as often as desired. *Note: This is a single card image command.*

Command structure:

```
EXPAND  NAMBLK
```

P2	(A)	NAMBLK	The name of the **DEFINE** block associated with the CEDRIC commands to be inserted and executed.
P3-10		UNUSED	++++++++++++++++++++
			(DEFAULT = The user is permitted to have one unnamed **DEFINE** block. In that case, NAMBLK can be blank.

FIELDSET—Divides all exisiting edit fields into two groups. This command is particularly useful when generating displays of similar fields, since it can eliminate the need for explicit reference to individual fields in the display stack. Whenever a new edit file is established using either the **READVOL** or **SYNTHES** command, all the fields are designated as primary Any fields subsequently transferred in or generated by edit commands are designated as secondary. Taken together, the primary and secondary field groups comprise the set of ALL fields. Any fields not included in the primary group are automatically classified as secondary, and vice versa. These groupings are mutually exclusive. **FIELDSET** is a multi-card image command whenever 8 or more fields are to be designated as comprising a group. *Note: This command must be terminated with a blank field name. Therefore, if any card image including the* **FIELDSET** *command line has a name as parameter (P10), the next card image* **must** *be blank.*

Command structure:

```
FIELDSETNAMGRP  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD
        NINCLD  NINCLD  NINCLD
```

P2	(F)	NAMGRP	Group name to be defined: `PRI` primary fields to be specified; `SEC` secondary fields to be specified. (DEFAULT = `PRI`)
P3-10	(A)	NINCLD	Names of the fields to be included in the group. Whenever 8 or more fields are to be specified their names must appear on subsequent card images in the following format:

FIELDSET –STACK of EXTRA CARD IMAGES–

P2-10	(A)	NINCLD	Remaining field names. Include as many EXTRA card images as needed. They will be processed until a “blank” field is encountered. If exactly 8 fields are specified, a “blank” card image *must* be provided in order to terminate the processing of field names.

FILTER—Performs two- or three-dimensional spatial filtering of any edit field using any one of 8 available techniques. The uniform (all data are equally-weighted), forward Hanning (weights in each dimension are 1/4, 1/2, 1/4), inverse Hanning (weights are designed to give a nearly-uniform response over the low-pass bandwidth of the Hanning filter), and two-way Hanning (weights are the same as forward, followed by inverse) filtering schemes are done *only* over local (3×3) regions. This means that the first zero in the filter response function is at 2Δ, where Δ is the grid spacing. The linear least-squares and Leise methods allow the user to select larger regions for filtering, thereby limiting the output to wavelengths longer than those from the uniform and Hanning filters. The Leise filter is described in NOAA Tech. Memo. ERL WPL–82 written by J.A. Leise entitled "A Multidimensional Scale-Telescoped Filter and Data Extension Package." This method is analagous to two-way Hanning—with NSTEP = 1.0, virtually the same as two-way Hanning with one pass at the grid spacing; with NSTEP = 2.0, two-way Hanning, once at the grid spacing and once at twice the grid spacing; with NSTEP = 3.0, two-way Hanning, once at the grid spacing, once at twice the grid spacing, and once at three times the grid spacing and so forth. Therefore, the first zeroes for the Leise multi-step filter are: 2Δ, 4Δ, 8Δ... for NSTEP =1.0, 2.0, 3.0 .... For all the methods, only those locations within the designated geometric window receive a filtered result. *Note: This is a single card image command.*

Command structure:

```
FILTER  NAMOUF  NAMINF  METHOD  NSTEP                               IFIXAX  IWINDO
```

P2	(A)	NAMOUF	Name of the output field containing the filtered result. (DEFAULT = The input field will be updated.)
P3	(A)	NAMINF	Name of the input field to be filtered.
P4	(A)	METHOD	Filtering method: `UNI` Uniform weighting of all values; `HAN` Hanning filtering will be done; `INV` Inverse Hanning filtering will be done; `TWO` Hanning followed by Inverse Hanning filtering will be done; `L1D` Leise one-dimensional filtering will be done; `LEI` Leise two-dimensional filtering will be done; `L3D` Leise three-dimensional filtering will be done; `LEA` Linear least-squares two-dimensional filtering will be done; `LT3` Linear least-squares three-dimensional filtering will be done.
P5	(F)	NSTEP	Only required for Leise and Least Squares filtering. If METHOD = `L1D,LEI,L3D`: The number of filter steps as defined by Leise; (DEFAULT = `1.0`) If METHOD = `LEA` or `LT3`: Specifies the radius in grid steps (number of grid steps outward from the central grid point) over which filtering will be done. (DEFAULT = `1.0`)
P6-8		UNUSED	++++++++++++++++++++
P9	(A)	IFIXAX	Axis to be fixed for two-dimensional filtering (or direction of filtering for one-dimensional filtering): `Z` z-axis fixed, operate in xy-planes(filter in z direction for L1D); `Y` y-axis fixed, operate in xz-planes(filter in y direction for L1D); `X` x-axis fixed, operate in yz-planes(filter in x direction for L1D). (DEFAULT = `Z`)

P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

FIXIDS—Alters information in the 510 word header of the edit volume that is currently active. This volume header consists of 16-bit words. Refer to the CEDRIC 16–bit VOLUME HEADER DESCRIPTION in **Appendix D** for a detailed layout of this data structure. As many as 4 words may be altered by a single **FIXIDS command**. *Note: This is a single card image command.*

Command structure:

```
FIXIDS  IWORD   IVALUE  IWORD   IVALUE  IWORD   IVALUE  IWORD   IVALUE
```

P2	(F)	IWORD	Position of word to be altered.
P3	(H)	IVALUE	New contents of word specified by P2.
P4	(F)	IWORD	Position of word to be altered.
P5	(H)	IVALUE	New contents of word specified by P4.
P6	(F)	IWORD	Position of word to be altered.
P7	(H)	IVALUE	New contents of word specified by P6.
P8	(F)	IWORD	Number of a word to be altered.
P9	(H)	IVALUE	New contents of word specified by P8.
P10		UNUSED	+++++++++++++++++++

FLTERTH—Enables or disables a flat earth mode. If enabled, coordinate transformations to three-dimensional Cartesian will not use curvature of the earth corrections in calculation of the height (Z). These corrections only affect the **ELEV** function within the **FUNCTION** command and the **REMAP** command when remapping from coplane or constant elevation angle coordinates to Cartesian. In CEDRIC, flat earth mode is disabled by default and is only enabled through the use of this command. *Note: This is a single card image command.*

Command structure:

```
FLTERTH OPTION
```

P2	(A)	OPTION	`ON` indicates that flat earth mode should be enabled. `OFF` indicates that flat earth mode should be disabled. (DEFAULT = `OFF`)
P3-10		UNUSED	+++++++++++++++++++

FUNCTION—Algebraic manipulation of multi-dimensional Cartesian fields. This command is a stack operation that allows the user to chain algebraic primitives together, thereby producing elaborate formulae without the unnecessary generation of superfluous edit fields. The stack consists of additional card images containing the primitive algebraic operations to be executed in the order in which they appear in the stack. A maximum of 30 primitives is permitted in a single stack. The algebraic primitives are defined in **Appendix B**, and they may be referenced either by name or by number. Only those locations within the designated window will contain the results of any computations. The stack *must* be terminated by an **END** command.

Fields within the **FUNCTION** stack may be any one of three types:

> **PERMANENT**—field either already exists or will be created as a permanent edit field upon completion of the command;
>
> **TEMPORARY**—field will exist only for the duration of the command;
>
> **CONSTANT PLANE**—after initialization the elements of this two-dimensional field are retained from one plane to the next. This field will exist only for the duration of the command. The constant plane provides the user with a CEDRIC internal field in which results can be accumulated as the selected operation proceeds in the specified direction. For example, if the user wanted to find the maximum value of a field within each vertical column of the grid, the first level of the field could be stored in this constant plane. Then, proceed by taking the maximum of the values of the field and the constant plane and putting this value back in the constant plane field. As the stack executes, a permanent field also should be set to the constant plane values where they can be stored after completion of the **FUNCTION** command. See the illustrative example included below.

Command structure:

```
FUNCTIONIDMODE  DIREC                                                   IFIXAX  IWINDO
        NAMOUF  IFLTYP  NAMFUN  NAMF1   NAMF2   C1      C2      C3      C4
        NAMOUF  C       CONPLANEINFLD   CONSET
END
```

An illustrative example of the `FUNCTION` *command:*

```
DEFINE  MAXWIN
COMMENT
        Obtain the accumulated maximum value over several planes of FINP
           1) Initialize FACCUM(I,J) to first plane of TFINP(I,J,K)
           2) Then for increasing K (IFIXAX direction)
              FACCUM(I,J) = MAX[FACCUM(I,J),TFINP(I,J,K)]
              FMAX(I,J,K) = FACCUM(I,J)
        Each level of FMAX contains the maximum value of FINP to that level
COMEND
VALUE   ON
FUNCTIONNAME    1.0                                             IFIXAX  WINDOW
        FMIN    T       CON                     CMIN
        TFINP   T       ORELSE  FINP    FMIN
        FACCUM  C       CONPLANETFINP   1.0
        FACCUM  C       MAX     FACCUM  TFINP
        FMAX    P       LINEAR  FACCUM          1.0     0.0
END
VALUE   OFF
DEFEND  MAXWIN
```

P2 (A) IDMODE — Definition mode for referencing primitives:
- `NUMBER` referenced by number;
- `NAME` referenced by name.

(DEFAULT = `NAME`)

P3	(A)	DIREC	Direction of the calculations through the volume: `+1.0` positive direction through the planes; `-1.0` negative direction through the planes. (DEFAULT = `+1.0`)
P4-8		UNUSED	++++++++++++++++++++
P9	(A)	IFIXAX	Axis to be fixed for two-dimensional functions: `Z` z-axis fixed, operate in xy-planes; `Y` y-axis fixed, operate in xz-planes; `X` x-axis fixed, operate in yz-planes. (DEFAULT = `Z`)
P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

FUNCTION			–STACK of ADDITIONAL CARD IMAGES–
P2	(A)	NAMOUF	Name of the output field containing the results of this primitive operation.
P3	(A)	IFLTYP	Output field type: `T` temporary; `C` constant plane field that has been previously defined; `P` permanent (CEDRIC decides the scaling factor if the field is being newly created;) `P*nnnnnn` new permanent field being created where `nnnnnn` is a user-specified 6-digit floating point value which will be used to scale the information in the field before it is stored. This feature should only be exercised by experienced users. (DEFAULT = If the output field name was previously used, the output field type is the same as originally defined. If the output field has not been defined yet, the output field type is temporary.
P4	(H)	NAMFUN	Function specification. Depending upon the value of NAMFUN the remaining information on the card image can be interpreted in either of two ways:

`---NAMFUN (P4) from the list of primitives in Appendix B---`

			Any valid FUNCTION NAME or NUMBER interpreted in the context indicated by IDMODE will add a primitive algebraic operation to the **FUNCTION** stack; P5 thru P10 contain parameterization information.
P5	(A)	NAMF1	Name of the first input field, if needed.
P6	(A)	NAMF2	Name of the second input field, if needed.
P7	(H)	C1	Name of the first constant, if needed.
P8	(H)	C2	Name of the second constant, if needed.
P9	(H)	C3	Name of the third constant, if needed.

P10	(H)	C4	Name of the fourth constant, if needed. If any constant = `BAD`, it will be set to the missing data flag when the function is executed. Otherwise, the (DEFAULT = `0.0`)

`---NAMFUN (P4) set to CONPLANE---`

			Establishes a CONSTANT PLANE field; P5 and P6 contain initialization information.
P5	(A)	INIFLD	Name of the field with which to initialize the CONSTANT PLANE field `NAMOUF`. (DEFAULT = Do not initialize with an existing field; instead set to a constant, see P6).
P6	(F)	CONSET	If an existing field has been specified in `INIFLD`, then `CONSET` is the index number of the fixed plane to use for the initialization of `NAMOUF`. (DEFAULT=Use the first plane within the window starting in the direction of the computation) Otherwise, if `INIFLD` is blank, `CONSET` contains the constant value to which `NAMOUF` will be initialized. (DEFAULT = `0.0`)
P7-10		UNUSED	+++++++++++++++++++

END This **END** command terminates the **FUNCTION** command stack.

GRAPHICS—Generates plotted displays for a set of designated fields. This is a stack operation and *must* be terminated by an **END** command. If a field appears more than once in the same type of plot in a single stack, only the last specification will be used to display that field. For example, a field can be used in a vector plot and in a contour plot in a single stack, but it cannot be used in two contour plots. If the user desires two differently specified contour plots of the same field, this would be done in two different **GRAPHICS** stacks. Fields will be displayed according to their order within the CEDRIC program and not according to their order in the stack.

The **GRAPHICS** command stack consists of additional card images that allow the user to specify the types of graphs and the fields to be graphed, along with associated display parameterizations. The command stack can consist of:

CONTOUR—produces ordinary contour lines or color area-fill;

VECTOR—produces a plot of two fields as a vector, with direction and length proportional to magnitude, or as streamlines. These plots can be overlaid on `CONTOUR` plots;

SCATTER—produces a scatter plot of one field against another, along with the parameters of a regression analysis (if selected);

ALPHA—produces an alphanumerically-coded display;

AIRTRCK—produces a plot of an aircraft track read in previously with the **READAIR** command.

STALOC—produces a plot of mesonet stations (or other landmarks) read in with the **READSTA** command.

DIGITAL—produces a digitally-coded display;

LABELS—controls the amount of labeling that appears on plots;

OVERLAY—produces an overlaid plot. This type of plot allows the user to combine any `CONTOUR` plot type which has been defined in the stack with a secondary overlay image. `OVERLAY` plots are done on a plotting area that is smaller than other plot types in order to accomodate the additional labeling that is required. The overlay image must be defined in the stack and can only be one of the following plot types: `CONTOUR, ALPHA` or `DIGITAL`.

Command structure:

```
GRAPHICSIX*IY*IZIFIXAX  SKIP    ITITLEITITLEITITLEITITLEITITLEITITLE    IWINDO
        CONTOUR NAMFLD  ICONTP  IOPTNS  DIG THK ICLDEF  CLMIN   CLMAX   CLINC
        VECTOR  NAMHOR  NAMVER  IVPTYP  IVSPAC  VMS     VKM     IVHEAD  COLOR
        SCATTER NAMHOR  XBEG    XEND    NAMVER  YBEG    YEND    DEGFIT  XCOL
        ALPHA   NMSPEC  SCLFAC  ZREF    WIDTH   NLFT    NRGT    ISPAC
        AIRTRCK                 IOPTNS  CMARKS  VMS     VKM     VFREQ   COLOR
        STALOC  IPLTPAR
        DIGITAL NMSPEC  SCLFAC  NDIG    ISPAC
        LABELS  LABFLG
        OVERLAY IFPRIM  IFOVLY  IOPTYP  IDASHP  IDASHO  PRICOL  OVLCOL
END
```

P2 (H)	IX*IY*IZ	Graphics skipping factor. This is a special mnemonic to produce graphical displays at integer multiples of the actual grid of the *current* edit volume. Each of `IX, IY,` and `IZ` are integer skipping factors (I2 format) that are used when passing through the fields to construct inputs to the graphical routines. For example, `02*02*01` will produce contour, color area-fill, vector, ... plots at twice the edit volume xy-grid spacing without degrading the resolution for other commands such as **FUNCTION**. If any of `IX, IY,` or `IZ` are set, this value takes precedence over SKIP (P4). However, the vector spacing (IVSPAC, P6 if P2 is `VECTOR`) will be additionally done, since it is applied to the grid spacing implied by use of `IX*IY*IZ`. (DEFAULT = `01*01*01`)

P3 (A) IFIXAX — Axis to be fixed for two-dimensional displays with optional Z-axis stretching whenever X or Y are fixed:

`Z`	z-axis fixed, display xy-planes;
`Y`	y-axis fixed, display xz-planes;
`X`	x-axis fixed, display yz-planes;

P3 (A) IFIXAX — Axis to be fixed continued.

`Y*nnnnnn` y-axis fixed, display xz-planes with Z-axis stretching. The stretching factor `nnnnnn` is a 6 digit floating point value specifying the desired ratio of 1 km along the vertical Z-axis to 1 km along the horizontal X–axis;

`X*nnnnnn` x-axis fixed, display yz-planes with Z-axis stretching. The stretching factor `nnnnnn` is a 6 digit floating point value specifying the desired ratio of 1 km along the vertical Z-axis to 1 km along the horizontal Y–axis;

(DEFAULT = `Z`)

P4 (F) SKIP — Increment between display levels. A display will be generated every SKIP levels.
(DEFAULT = `1.0`)

P5-9 (A) ITITLE — User supplied titling information; appears at top of each plot generated. (40 Character maximum.)
(DEFAULT = No user supplied titling.)

P10 (A) IWINDO — Windowing specification for the display:

`WINDOW`	display will be confined to the current user-set window;
`FULL`	display will be generated over the full (X,Y,Z) grid.

(DEFAULT = `FULL`)

GRAPHICS –STACK of ADDITIONAL CARD IMAGES–

P2 (A) IPLTYP — Plot type specifier. If the 8th character of IPLTYP contains the letter `D`, the plots specified by that card image will be DEFINED but not generated. This feature allows the user to define graphics for subsequent use in OVERLAYs without having to produce those graphics as separate plots. Depending on the plot type designated by IPLTYP the remaining information on the card image will be interpreted accordingly:

`---IPLTYP (P2) set to CONTOUR---P3-P10 contain parameterization---`

P3 (A) NAMFLD — Contour field name or one of the following codes designating a predetermined group of fields:

`PRI`	the primary field group will be displayed;
`SEC`	the secondary field group will be displayed;
`ALL`	every field will be displayed.

See the **FIELDSET** command for instructions on how to set primary and secondary field groups.
(DEFAULT = `PRI`)

P4 (A) ICONTP

Type of contouring or color area-filling desired, with the maximum number of contour levels in parentheses:

`CONT`	contour lines only (61 levels)
`HAFT`	halftone shading only (15 levels)
`CHAFT`	contour lines with halftone shading
`FALL`	area-fill using all colors, no contour lines (61 levels);
`FHOT`	area-fill using "hot" colors, no contour lines (31 levels);
`FCOLD`	area-fill using "cold" colors, no contour lines (31 levels);
`CFALL`	area-fill using all colors, with contour lines (61 levels);
`CFHOT`	area-fill using "hot" colors, with contour lines (31 levels);
`CFCOLD`	area-fill using "cold" colors, with contour lines (31 levels);

`GREYS` area-fill using shades of grey, no contour lines (61 levels). The shades of grey increase monotonically in intensity from the lowest contour level to the highest.

`CGREYS` area-fill using shades of grey, with contour lines (61 levels). The shades of grey increase monotonically in intensity from the lowest contour level to the highest.

`CGREYS2` area-fill using shades of grey, with contour lines. The middle bin is blank and a three-peat pattern moving in both directions from the middle bin is used. (61 levels).

`CHAFT2` area-fill using halftones, with contour lines. The middle bin is blank and a three-peat pattern moving in both directions from the middle bin is used. (61 levels).

`CGREYS3` area-fill using shades of grey, with contour lines. The shades of grey increase monotonically in intensity from the lowest contour level to the highest. The middle bin, however, is blank. (61 levels).

`CGREYS4` area-fill using shades of grey, with contour lines. A three-peat pattern starting from the lowest contour level is used. (61 levels).

(DEFAULT = `CONT`)

P5 (A) IOPTNS

Contouring options. IOPTNS is a alphanumeric, left-justified pattern of single digit numbers most of which either turn **OFF** (0) or **ON** (1) a particular plotting feature. There are 8 features over which the user has control. In order from left to right in the 8-digit pattern, they are:

1—Label the contour lines [DEFAULT is 0 (OFF)];

2—Mark relative highs and lows [DEFAULT is 0 (OFF), 1 means mark "many" highs and lows, 2 means mark "fewer" up to 9 which means to mark the fewest number of highs and lows. 4 is a reasonable value for marking the most prominent highs and lows.];

3—Dashed line pattern and color:

[The first color in parentheses following the type of line is the color of the contour lines when the contour type (P4) is `CONT, BOTH, CFALL, CFHOT`, or `CFCOLD`. The second color in parentheses is the color of the contour lines when the contour type is `CGREYS`].

0—Positive is solid and negative is dashed (white, white);
1—Alternating pattern of three dashed types (white, white);
2—Positive is solid and negative is dashed (black, black);
3—Alternating pattern of three dashed types (black, black);
4—Positive is solid and negative is dashed (grey, shades of grey);

5—Alternating pattern of three dashed types (grey, shades of grey);
6—Positive is solid and negative is dashed (blue, grey);
7—Alternating pattern of three dashed types (blue, grey);
8—Positive is solid and negative is dashed (yellow, grey);
9—Alternating pattern of three dashed types (yellow, grey);
[DEFAULT = 1]

4—Stop contours at missing data locations [DEFAULT = 0 (OFF)];
5—Indicate maximum value on the plot [DEFAULT = 1 (ON)];
6—Digitize the field on the plot background [DEFAULT = 0 (OFF)];
7—Overlay vector or airtrck plots;
0—No overlay;
1—Overlay the last of the vector plots specified in the stack with this contour plot;
2—Overlay the last of the airtrck plots specified in the stack with this contour plot;
3—Overlay the last vector and airtrck plots specified in the stack with this contour plot;
4—Overlay the last station location (staloc) plot specified in the stack with this contour plot;
5—Overlay the last staloc and vector plots;
6—Overlay the last staloc and airtrck plots;
7—Overlay the last staloc, vector, and airtrck plots;

8—Overlay map background/landmarks on this plot according to the value of this digit:
0—No overlay;
1—Map background;
2—Landmarks in the header;
3—Both (1) and (2) [DEFAULT = 0 (OFF)].

(DEFAULT = `00101000`)

P6	(F)	DIG THK	Scaling factor and contour line thickness. The DIG portion is used only when digitizing field values on the plot background. Values will be multiplied by this number before they are displayed. (DEFAULT = The program decides.) The THK portion is used whenever contour lines are drawn on the plot. A value of 1.0 (the DEFAULT) means to use the default line thickness. A higher number means to use thicker lines. 2.0 will give noticably thicker lines. Anything less than 0.5 will cause the lines to disappear. The DIG scaling factor uses the first 4 digits of this parameter and the THK thickness factor uses the last 4 digits. Both are floating point values.
P7	(A)	ICLDEF	Contour line specification mode: `UNI` uniform contour levels will be specified by P8-10; `NON` P8-10 are ignored and non-uniform contour lines will be specified by P2-10 of **EXTRA** card images. (DEFAULT = `UNI`)
P8	(F)	CLMIN	Beginning contour level when ICLDEF=`UNI`; (DEFAULT = `0.0`).
P9	(F)	CLMAX	Ending contour level when ICLDEF=`UNI`; (DEFAULT = `60.0`).
P10	(F)	CLINC	Increment between levels when ICLDEF=`UNI`; (DEFAULT = `10.0`).

`---IPLTYP (P2) set to CONTOUR---EXTRA card images to set non-uniform contour levels---`

P2-10 (F)	User supplied contour levels. Used when ICLDEF=`NON`. These **EXTRA** card images are processed until a "blank" field is encountered.

`---IPLTYP (P2) set to VECTOR---P3-P10 contain parameterization---`

P3 (A) NAMHOR — Field name for the vector component along the current I-axis.

P4 (A) NAMVER — Field name for the vector component along the current J-axis.

P5 (A) IVPTYP — Type of vector plot:
`ARROW` arrows will be drawn according to specifications in P6-P9;
`STREAM` streamlines will be generated according to the value specified by P6 (P7-P9 are ignored).
(DEFAULT = `ARROW`)

P6 (F) IVSPAC — Number of grid spaces between vectors:
If IVPTYP = `ARROW`:
–1 interleaved (odd border points);
–2 interleaved (even border points);
>0 every IVSPAC vector will be plotted.
(DEFAULT = `-1.0`)
If IVPTYp = `STREAM`:
>0 a streamline will be initiated every IVSPAC grid locations;
(DEFAULT = `2.0`)

P7 (F) VMS — Reference speed (M/S); only if IVPTYP =`ARROW`. (DEFAULT = `2.0`)

P8 (F) VKM — Reference distance (KM); only if IVPTYP =`ARROW`. (DEFAULT = `1.0`) Arrow lengths are scaled so that VMS (M/S) is VKM (KM) in length on the plot.

P9 (A) IVHEAD — Arrowhead type; only if IVPTYP =`ARROW`:
`FIX` arrow heads will be a fixed size;
`PROPnnnn` arrow heads will be proportional to the arrow lengths, and if `nnnn` is between 0.01 and 5.0, scaled by `nnnn`. (`nnnn` is in F4.2 format);
(DEFAULT = `FIX`)

P10 (A) COLOR — Color of vectors. The options are `WHITE, GREY, PURPLE, BLUE, GREEN, YELLOW, BROWN, MAGENTA, BLACK`, and `RED`. **NOTE:** when vectors are overlayed on contour plots filled with shades of grey only `WHITE, GREY,` and `BLACK` are possible choices for vector color.
(DEFAULT = `WHITE`)

`---IPLTYP (P2) set to SCATTER---P3-P10 contain parameterization---`

P3 (A) NAMHOR — Field name for horizontal axis (independent variable).

P4 (F) XBEG — Left endpoint of the horizontal axis. (DEFAULT = `-50.0`)

P5 (F) XEND — Right endpoint of the horizontal axis. (DEFAULT = `50.0`)

P6 (A) NAMVER — Field for vertical axis (dependent variable).

P7 (F) YBEG — Left endpoint of the vertical axis. (DEFAULT = `-50.0`)

P8 (F) YEND — Right endpoint of the vertical axis. (DEFAULT = `50.0`)

P9	(F)	DEGFIT	Degree of the polynomial least-squares fit; the vertical axis variable computed as a function of the horizontal axis variable. (DEFAULT = `0.0`). No curve fitting will be performed.
P10	(A)	XCOL	The first character `X` is the keyboard character to be used in scatter plots, and `COL` is the color to be used for the character plotted. The color options are: `WHITE, GREY, PURPLE, BLUE, GREEN, YELLOW, BROWN, MAGENTA`, and `RED`. For example, the string `+BLUE` would plot `+` characters in blue. (DEFAULT = `+WHITE`), plus signs in white.

`---IPLTYP (P2) set to ALPHA---P3-P9 contain parameterization---`

P3	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` the primary field group will be displayed; `SEC` the secondary field group will be displayed; `ALL` every field will be displayed. (DEFAULT = `PRI`)
P4	(F)	SCLFAC	Scaling factor. Field values will be multiplied by this number before they are encoded. (DEFAULT = `1.0`)
P5	(F)	ZREF	Reference value for display. All field values less than ZREF will be encoded as lower-case letters; all field values greater than ZREF will be represented as upper-case. Values beyond the boundaries set by the user will be represented by either plus (+) or minus (-), depending upon their sign. Missing data values will be represented by a period (.) (DEFAULT = `0.0`)
P6	(F)	WIDTH	Interval width for the individual bins. (DEFAULT = `5.0`)
P7	(F)	NLFT	Number of bins to the left of reference value (0–26) (DEFAULT = `0.0`)
P8	(F)	NRGT	Number of bins to the right of reference value (0–26) (DEFAULT = `0.0`)
P9	(H)	ISPAC	ALPHA symbol spacing and background specification. This information is supplied in the following 8-character format: `M/nnnnnn` where `M` and `nnnnnn` are supplied by the user and separated by a slash (/). The `M` is a single letter code that designates how missing values are to be represented in the display: `P` a period (.) will be digitized at missing data locations; `B` missing data locations will remain blank. The quantity `nnnnnn` is a 6-digit floating point number designating the symbol spacing: `-1.0` interleaved (odd border points); `-2.0` interleaved (even border points); `>0.0` every `nnnnnn`'th symbol will be plotted. The combined (DEFAULT = `P/-1.0`). Interleave odd with missing data represented by periods. Note: `M` and `nnnnnn` are separately DEFAULTed.
P10		UNUSED	+++++++++++++++++++

---IPLTYP (P2) set to AIRTRCK---P3-P6 contain parameterization---

Note that the program automatically searches for the wind components that correspond to the fixed axis in effect. For example, if z is the fixed axis (x-y planes), the u v components, if available and the winds plotting option below is turned on, will be plotted.

P3-P4		UNUSED	++++++++++++++++++++
P5	(A)	IOPTNS	Plotting options. IOPTNS is an alphanumeric, left-justified pattern of single digit numbers most of which either turn off or on a particular plotting feature. There are 6 features over which the user has control. In order from left to right in the 8-digit pattern, they are: 1—Dash pattern option. If 0, the pattern will be dashed if the track position is less than the current level and solid if the track is greater than or equal to the current level. If 1, the pattern will always be dashed. If 2, the pattern will always be solid. 2—Flag to indicate if beginning and ending times of the track are to be plotted. If 1, the times will be plotted on the frame. If 0, they will not be plotted. 3—Flag to indicate if vector winds (if available) are to be plotted. If 1, the winds will be plotted. If 0, they will not be. 4—Indicates what kind of symbol to plot for major marks on the airtrack (see P6 for more details). 0–for a circle with a plus sign in the middle 1–for a plus sign 2–for a triangle 3–for a circle with a dot in the middle 4–for a box 5—Indicates what kind of symbol to plot for minor marks on the airtrack (see P6 for more details). 0–for a plus sign 1–for a circle with a plus sign in the middle 2–for a triangle 3–for a circle with a dot in the middle 4–for a box 6—Multiplication factor for aircraft-track plot limits. Track is plotted if the aircraft is within ± multfac*delta km of the current display level, where delta is grid spacing. If this parameter is not specified or if there is only one level, the entire track is plotted. (DEFAULT = 011000)
P6	(F)	MAJ MIN	Specifies how often major and minor marks should be placed along the track. The first 4 spaces of this parameter are how often major marks should be placed and the second 4 are how often minor marks should be placed in seconds. For example, '36001800' would indicate that a major mark is to be placed every hour (3600 seconds) and a minor mark every half hour. If no marks are desired for one or both, leave one or both parts of this parameter blank or set to zero. For example, '36000000' indicates that a major mark should be placed every hour; no minor marks are to to placed on the track. The symbols used are specified with P5 of this command. Note that these values must be integer multiples of the resolution at which the data is read in.
P7	(F)	VMS	Reference speed (M/S). (DEFAULT=None.)

P8	(F)	VKM	Reference distance (KM). Arrow lengths are scaled so that VMS (M/S) is VKM (KM) in length on the plot.
P9	(F)	VFREQ	Specifies how often vectors are to drawn along track in seconds. For example, '3600' would indicate that vectors are to be drawn every hour. Vector wind fields must exist and the third digit of P5 (above) must be turned on. (DEFAULT=no vectors).
P10	(A)	COLOR	Color of airtrack. The options are `WHITE, GREY, PURPLE, BLUE, GREEN, YELLOW, BROWN, MAGENTA`, and `RED`. NOTE: when vectors are overlayed on contour plots filles with shades of grey, only `WHITE, GREY,` and `BLACK` are options. (DEFAULT = `WHITE`)

`---IPLTYP (P2) set to STALOC---P3-P6 contain parameterization---`

P3	(F)	IPLTPAR	Plotting parameter that controls how the stations are marked on the plots. `1.0`–plot symbol only `2.0`–plot name only `3.0`–plot symbol and name `4.0`–plot symbol only in a box `5.0`–plot name only in a box `6.0`–plot symbol and name in a box (DEFAULT = `NONE. User must specify.`)
P4-P10		UNUSED	+++++++++++++++++++++

`---IPLTYP (P2) set to LABELS---P3 contains parameterization---`

P3	(F)	LABFLG	Specifies how much labeling is to appear on all plots that follow this command. This command is useful for preparing plots for publication. Currently, the only values supported are 9.0 (full labeling) and 1.0 (minimal labeling). (DEFAULT = `9.0`)
P4-P10		UNUSED	+++++++++++++++++++

`---IPLTYP (P2) set to DIGITAL---P3-P6 contain parameterization---`

P3	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` the primary field group will be displayed; `SEC` the secondary field group will be displayed; `ALL` every field will be displayed. (DEFAULT = `PRI`)
P4	(F)	SCLFAC	Scaling factor. Field values will be multiplied by this number before they are encoded. (DEFAULT = `1.0`)
P5	(F)	NDIG	Maximum number of digits allocated for each value. Values that cannot be plotted using the number of digits allocated will be represented as either plus (+) or minus (-), depending on their sign. Missing data values will be represented by a period (.) (DEFAULT = `3.0`)
P6	(H)	ISPAC	DIGITAL value spacing and background specification. This information is supplied in the following 8-character format: `M/nnnnnn` where `M` and `nnnnnn` are supplied by the user and separated by a slash (/). The `M` is a single letter code that designates how missing values are to be represented in the display: `P` a period (.) will be digitized at missing data locations; `B` missing data locations will remain blank. The quantity `nnnnnn` is a 6-digit floating point number designating the symbol spacing: `-1.0` interleaved (odd border points); `-2.0` interleaved (even border points); `>0.0` every `nnnnnn`'th symbol will be plotted. The combined (DEFAULT = `P/-1.0`). Interleave odd with missing data represented by periods. Note: `M` and `nnnnnn` are separately DEFAULTed.
P7-10		UNUSED	+++++++++++++++++++

`---IPLTYP (P2) set to OVERLAY---P3-P9 contain parameterization---`

P3	(A)	IFPRIM	Primary field name. This must be an actual edit field name. A `CONTOUR` plot corresponding to this field name must be defined by some means elsewhere in the stack.
P4	(A)	IFOVLY	Overlay field name. This must be an actual edit field name. A `CONTOUR`, `ALPHA` or `DIGITAL` plot corresponding to this field name must be defined by some means elsewhere in the stack.
P5	(A)	IOPTYP	Overlay plot type: `CONTOUR` contour or color area-fill plot; `ALPHA` alphanumeric plot; `DIGITAL` digital or numeric plot; (DEFAULT = `CONTOUR`)
P6	(A)	IDASHP	Dashline pattern to be used for any contour lines appearing in `CONTOUR` plot of the primary field: `S` solid lines; `D` short dashed lines; `L` long dashed lines; `N` long dashed negative and solid positive lines; `M` multiple pattern dash-lines; (DEFAULT = `S`)
P7	(A)	IDASHO	Dashline pattern to be used for any contour lines appearing in plot of the overlay field. This parameter is only necessary when the overlay field is a `CONTOUR` plot. See the above description of IDASHP for the list of valid dashline codes. (DEFAULT = `L`)
P8	(A)	PRICOL	Color used for any contour lines in the primary field. The choices are: `BLACK, GREY, WHITE, BLUE, GREYS` (for varying shades of grey), and `YELLOW`. *NOTE:* When the primary plot is contour lines filled with shades of grey, only `WHITE, BLACK, GREY,` and `GREYS` are possible choices for PRICOL. (DEFAULT = `WHITE`)
P9	(A)	OVLCOL	Color used for any contour lines in the overlay field. The choices are: `BLACK, GREY, WHITE, BLUE, and YELLOW`. *NOTE:* When the primary plot is contour lines filled with shades of grey, only `WHITE, BLACK, GREY,` and `GREYS` are possible choices for `OVLCOL`. (DEFAULT = `WHITE`)
P10		UNUSED	+++++++++++++++++++

END This **END** command terminates the **GRAPHICS** command stack.

HISTO—Generates a histogram display for a set of designated fields. This is a stack operation and *must* be terminated by an **END** command. The command stack consists of additional card images that allow the user to specify the fields to be examined along with associated display parameterization. If a field is referred to more than once, the last specification will be used to display that field. Fields will be displayed in the order they are listed with this command. The maximum number of bins for any display is `1000`. Statistics such as mean and standard deviation will be displayed at the top of the plot. These statistics are calculated over the whole distribution, not just the range of bins selected.

Command structure:

```
HISTO   IDEST   IFIXAX  SKIP    ITYP    TTITLETTITLETTITLETTITLETTITL   IWINDO
        NMSPEC  SCLFAC  HMIN    HMAX    WIDTH   MEDIAN  COLOR   PERC    DISOPT
END
```

P2 (A) IDEST — Destination of the display:
All histograms will be placed in the graphics metacode file. There are currently no other valid destinations. Therefore, P2 can simply be left blank.

P3 (A) IFIXAX — Axis to be fixed for displays:
- `Z` z-axis fixed, construct histogram from xy-planes;
- `Y` y-axis fixed, construct histogram from xz-planes;
- `X` x-axis fixed, construct histogram from yz-planes.

(DEFAULT = `Z`)

P4 (F) SKIP — Increment between display levels. Histograms will be generated every SKIP levels.
(DEFAULT = `1.0`)

P5 (A) ITYP — Type of HISTO display to produce for each field:
- `PLANE` displays will be generated by plane;
- `VOLUME` displays will be generated by volume (all points within the windowed region);
- `BOTH` displays will be generated by plane and by volume.

(DEFAULT = `BOTH`)

P6-9 TTITLE — User-supplied additional titling for histograms.
(DEFAULT=none)

P10 (A) IWINDO — Windowing specification for the display:
- `WINDOW` display will be confined to the current user-set window;
- `FULL` display will be generated over the full (X,Y,Z) grid.

(DEFAULT = `FULL`)

HISTO –STACK of ADDITIONAL CARD IMAGES–

P2 (A) NMSPEC — Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields:
- `PRI` only the primary field group will be displayed;
- `SEC` only the secondary field group will be displayed;
- `ALL` every field will be displayed.

See the **FIELDSET** command for instructions on how to set primary and secondary field groups.
(DEFAULT = `PRI`)

P3	(F)	SCLFAC	Scaling factor. Field values will be multiplied by this number before they are placed in bins. (DEFAULT = `1.0`)
P4	(F)	HMIN	Left most midpoint of the frequency distribution. (DEFAULT = `0.0`)
P5	(F)	HMAX	Right most midpoint of the frequency distribution. (DEFAULT = `100.0`)
P6	(F)	WIDTH	Interval width for individual bins. (DEFAULT = `5.0`)
P7	(A)	MEDIAN	Median flag. `ON` a line indicating the median of the distribution will be drawn in the plot. The median is calculated over the whole distribution, not just within the range of bins that selected. (DEFAULT = `OFF`)
P8	(A)	COLOR	Color of histogram bars. The options are `NONE (don't fill) WHITE, GREY, PURPLE, BLUE, GREEN, YELLOW, BROWN, MAGENTA`, and `RED`. (DEFAULT = `NONE`)
P9	(F)	PERC	Highest percent to be drawn for any one bin. Any bin whose percent of total exceeds this number will be drawn up to this value, and then several asterisks will be placed on top of the bar to indicate that the percent within this bin actually goes higher. (DEFAULT = `100.0`)
P10	(A)	YAXIS	Controls y axis type: `LINEAR` axis will be linear. `LOG` axis will be logarithmic. (DEFAULT = `LINEAR`)

END This **END** command terminates the **HISTO** command stack.

INTEGR—Vertical integration of an existing edit field. This command is used mostly for the derivation of vertical air motions, with density weighting of the appropriate fields. See **Appendix F** for details on integrating the mass continuity equation. An empirical density weighting function is built into the command and is defined below. If a different form of density weighting is desired, the input field should be premultiplied using the facilities in the **FUNCTION** command, and the integration can be invoked with the internal density weighting switched off. The integration can be performed in Cartesian space or coplane space. CEDRIC will obtain the coordinate system information from the volume header, or the user can force the coordinate system to be used with the **COORD** command. There are three methods of integration: 1) upward from the lower boundary conditions, 2) downward from the upper boundary conditions, and 3) a variational scheme within each vertical column that requires both upper and lower boundary conditions. *Note: This is a single card image command.*

Command structure:

```
INTEGR  NAMOUF  NAMINF  INTYPE  METHBC  INIVAL  NAMTOP          ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  U*nnnnnnMETHBC  INIVAL  NAMTOP          ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  D*nnnnnnMETHBC  INIVAL  NAMTOP          ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  V*nnnnnnMETHUP  INIVUP  METHDN  INIVDN  ZLOWER  ZUPPER
```

P2 (A) NAMOUF — Name of the field for integration output.

P3 (A) NAMINF — Name of the existing field to be integrated.

P4 (A) INTYPE — Type of integration to be performed. This parameter is in the form `M*nnnnnn` where `M = U, D,` or `V` designates the method of integration and `nnnnnn` is the user supplied term in the density weighting function `RHO = EXP ( - Z * nnnnnn)`, where Z is the height in kilometers (MSL).
Whenever an asterisk appears after the method `M`, density weighting is assumed and `nnnnnn` is set to `0.1` if no value is supplied. Depending on the value of `INTYPE`, the remaining parameters are interpreted as described below. (DEFAULT = `V*0.1`)

`---INTYPE (P4) set to U*nnnnnn---P5-P10 contain parameterization---`

P5 (A) METHBC — Method for specifying the lower boundary condition for upward integration:

`CONST` initialize paths with a constant,
`Winit = Constant`;

`FRACT` initialize paths using a fraction of the field being integrated,
`Winit = NAMINF * FRACT * DIRECTION *` ΔZ,
where `DIRECTION` is (+1.0) for upward and (-1.0) for downward integrations and CEDRIC determines ΔZ.

`FIELD` initialize using the value from the same xy-location in another field,
`Winit = Value from another field.`

P6 (H) INIVAL — Value to be used when initializing using method designated by `METHBC`:
For `CONST` : Value of the constant. (DEFAULT = `0.0`)
For `FRACT` : Value of the fraction. (DEFAULT = `1.0`)
For `FIELD` : Name of the field to be used must be supplied.

P7 (A) NAMTOP — Name of a field in which topography information is to be saved from the integration. Topography is the height of first-occurrence (in the direction of integration) of the field being integrated.
(DEFAULT=Do not save any topography information.)

P8 UNUSED — +++++++++++++++++++

P9 (H) ZLOWER — Lowest Z-level to be included in the integration procedure:
`I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level;
`D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level;
(DEFAULT=lowest level in the volume).

P10 (H) ZUPPER — Highest Z-level to be included in the integration procedure:
`I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level;
`D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level;
(DEFAULT=highest level in the volume).

`---INTYPE (P4) set to D*nnnnnn---P5-P10 contain parameterization---`

P5 (A) METHBC — Boundary condition initialization method. (See the description above.)

P6 (H) INIVAL — Value to be used when initializing using the method designated by `METHBC`. (See the description above.)

P7 (A) NAMTOP — Name of a field in which topography information is to be saved from the integration. (See description above.)

P8 UNUSED — +++++++++++++++++++

P9 (H) ZLOWER — Lowest Z-level to be included in the integration procedure. (See the description above.)

P10 (H) ZUPPER — Highest Z-level to be included in the integration procedure. (See the description above.)

`---INTYPE (P4) set to V*nnnnnn---P5-P10 contain parameterization---`

P5 (A) METHUP — Method of specifying the lower boundary conditions. (See the description above.)

P6 (H) INIVUP — Value to be used when setting the lower boundary condition the method designated by `METBUP`. (See the description above.)

P7 (A) METHDN — Method of specifying the upper boundary conditions. (See the description above.)

P8 (H) INIVDN — Value to be used when setting the upper boundary conditions using the method designated by `METHDN`. (See the description above.)

P9 (H) ZLOWER — Lowest Z-level to be included in the integration procedure. (See the description above.)

P10 (H) ZUPPER — Highest Z-level to be included in the integration procedure. (See the description above.)

LAPLACE—Least-squares solution for pressure perturbations. Horizontal momentum equations with unknown pressure perturbations (p) on the left and "known" quantities (F and G) on the right:

$$\frac{\partial p}{\partial x} = -\rho_0 \frac{\mathcal{D}u}{\mathcal{D}t} + fr_x \equiv F$$
$$\frac{\partial p}{\partial y} = -\rho_0 \frac{\mathcal{D}v}{\mathcal{D}t} + fr_y \equiv G.$$

The above equations will have a solution for (p) if and only if

$$\frac{\partial F}{\partial y} = \frac{\partial G}{\partial x}.$$

Since the total derivatives of (u,v) on the right are determined inexactly, and the formulation for the friction terms (fr_x, fr_y) is imperfect, the system can be solved only in a least-squares sense. The solution to this variational problem leads to

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y},$$

where F and G are obtained from Doppler synthesized winds and some user-determined friction terms. The resulting partial differential equation is a Poisson equation for the pressure perturbations subject to Neumann boundary conditions. Results are computed by Z-level for every location in the (x,y,z) volume for which there are input values for the two "known" quantities on the right-hand side of the equation. This command can be used to generate an estimate of the pressure perturbation field at each level when values of $F = \partial p/\partial x$ and $G = \partial p/\partial y$ are provided as the input fields. **LAPLACE** is an implementation of the pressure retrieval technique described by Gal-Chen and Hane (1981) in their article in Vol. 12 of the NCAR Atmospheric Technology report entitled: "Retrieving Buoyancy and Pressure Fluctuations from Doppler Radar Observations". This command may consume MINUTES of cpu time. *Note: This is a single card image command.*

Command structure:
```
LAPLACE NAMOUF  DPDX     DPDY     RELEPS  MAXITR
```

P2	(A)	NAMOUF	Name for the resulting output pressure perturbation field.
P3	(A)	DPDX	Name of the input field (F) to be differentiated inside **LAPLACE** as the first term on the right hand side of the Poisson equation.
P4	(A)	DPDY	Name of the input field (G) to be differentiated inside **LAPLACE** as the second term on the right hand side of the Poisson equation.
P5	(F)	RELEPS	Epsilon value of the relative error to achieve in order to satisfy convergence. (DEFAULT = `0.001`)
P6	(F)	MAXITR	Maximum number of iterations to perform while convergence criteria (RELEPS) remains unsatisfied. (DEFAULT = `400.0`)
P7-10		UNUSED	++++++++++++++++++++

MASS2—This command computes a consistent set of wind component fields: (Unew, Vnew, Wnew), using the original synthesized winds (U,V) and the corresponding error terms (EWU,EWV) that are produced by invoking the two-radar approximation in the **SYNTHES**is command. The **MASS2** command expects the above fields to be present and assumes the above names. In order for this calculation to work properly, these input fields (generated by the synthesis) must not have been altered prior to invoking of this command, except perhaps by filtering and such. Under no circumstances should the (U,V) winds from the two-radar approximation already be altered in an attempt to account for the vertical motion (W) contamination. If (U, V, EWU or EWV) have been **RENAME**d, they must be **RENAME**d back to their original monikers before invoking this command. **MASS2** is a two-card image command that requires one **EXTRA** card image, but no **END** card image. See **Appendix F** for details on how the the mass continuity equation is solved for vertical motions. Either upward or downward integration will be done: no variational scheme has been implemented.

Command structure:

```
MASS2   UNEW    VNEW    WNEW    INTYPE  METHBC  INIVAL  CNCRIT  ZLOWER  ZUPPER
        ZINIT   NAMDBZ  ACON    BCON    CCON
```

P2 (A) UNEW — Name of the adjusted U-component output field.

P3 (A) VNEW — Name of the adjusted V-component output field.

P4 (A) WNEW — Name of the integrated W-component output field.

P5 (A) INTYPE — Type of integration to be performed. This parameter is in the form `M*nnnnnn` where `M` designates the method of integration and `nnnnnn` is the user supplied term in the density weighting function `RHO = EXP ( - Z * nnnnnn)`, where Z is the height in kilometers (MSL).
Whenever an asterisk appears after the method `M`, density weighting is assumed and `nnnnnn` is set to `0.1` if no value is supplied. Depending on the value of `INTYPE`, the remaining parameters are interpreted as described below.
(DEFAULT = `U*0.1`)

P6 (A) METHBC — Method for specifying the lower boundary condition for upward integration:

`CONST` initialize paths with a constant,
`Winit = Constant`;

`FRACT` initialize paths using a fraction of the field being integrated,
`Winit = NAMINF * FRACT * DIRECTION *` ΔZ,
where `DIRECTION` is (+1.0) for upward and (-1.0) for downward integrations and CEDRIC determines ΔZ.

`FIELD` initialize using a value at the same xy-location in another field,

`Winit = Value from a user-supplied field.`

P7 (H) INIVAL — Value to be used when initializing using method designated by `METHBC`:
For `CONST` : Value of the constant. (DEFAULT = `0.0`)
For `FRACT` : Value of the fraction. (DEFAULT = `1.0`)
For `FIELD` : A name of the field to be used *must* be supplied.

P8	(H)	CNCRIT	Criteria for determining procedure convergence at each level. This parameter is in the form `III.EEEE`, where `III` is the maximum number of iterations to perform at each level while attempting to satisfy the requirement that the mean difference between the vertical and horizontal convergence at each level be less than `EEEE`. The part `III` must be an integer whose absolute value is in the range `1-25`. If `III` is negative, CEDRIC will iterate exactly `ABS(III)` times, regardless of other convergence criteria. The part `EEEE` must be in the range `0.01-0.99`. (DEFAULT = `10.01`); however each part is DEFAULTed separately.
P9	(H)	ZLOWER	Lowest Z-level to be included in the integration procedure: `I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level; `D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level; (DEFAULT=lowest level in the volume).
P10	(H)	ZUPPER	Highest Z-level to be included in the integration procedure: `I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level; `D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level; (DEFAULT=highest level in the volume).

MASS2 –STACK of EXTRA CARD IMAGES–

P2	(H)	ZINIT	Level in the direction of the integration PAST which initialization will cease. This will be the last level at which the boundary conditions will be set. `I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level; `D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level; (DEFAULT = Continue to initialize at every level).
P3	(A)	NAMDBZ	Name of the radar reflectivity field to be used in the user-supplied Reflectivity — Fallspeed relationship: `VT = -A * (10.0**(0.1*DBZ*B)) * (RHO(0)/RHO(Z))**C`, where `DBZ` is the radar reflectivity in dBZ units; `RHO` is the air density (already specified); and `A, B, C` are supplied by the user in (P4-P6). (DEFAULT = Fallspeed will be set to `0.0`).
P4	(F)	ACON	Constant A (as described above). (DEFAULT = `1.5`)
P5	(F)	BCON	Constant B (as described above). (DEFAULT = `0.105`)
P6	(F)	CCON	Constant C (as described above). (DEFAULT = `0.4`)
P7-10		UNUSED	++++++++++++++++++++

PATCHER—Performs one- or two- or three-dimensional data filling (interpolation and extrapolation) or decimation of any scalar edit field. The methods are readily tailored to individual user-needs.

Data filling can be performed by using any one of three available techniques:

> **FILLCON**—Use a two-dimensional, linear, local-least-squares method with constraints. At each (I,J) grid location where a data value is missing (`BAD`) the program goes out one grid interval, accumulating the coefficients of a linear, least-squares fit in two-dimensions. After doing so, the user-specified constraints are checked: 1) are there good data values in each of `MINQAD` quadrants, a test of the degree that good data bounds the (I,J) missing-data grid location and 2) are there at least `MINPTS` good values involved, a test of sufficient numbers of good data values to make a reasonable fit If both tests are satisfied, the program solves the normal equations and moves on to the next (I,J) location having missing data. However, if either test fails the program searches out one more grid shell, and then repeats the tests. The search and testing procedure continues until either both tests succeed and a value is interpolated or the maximum number of steps outward (`MINPTS`) is reached. In the latter case, if the tests still fail, the missing data point remains flagged as `BAD`. By choosing four quadrants, the result is an interpolation and by choosing two quadrants, the result is an extrapolation. Three quadrants is an intermediate situation between interpolation and extrapolation. The method is piecewise-continuous, but no attempt is made to match the derivatives at the boundaries between adjacent regions.
>
> **FILLCON3**—Use a three-dimensional, linear, local-least-sqaures method with constraints. This method is nearly identical to the two-dimensional version except instead of testing on quadrants, the program tests on octants surrounding the missing data location.
>
> **FILLALL**—Globally fill all missing data points, using the Leise technique. The Leise method is described in NOAA Tech. Memo. ERL WPL–82 written by J.A. Leise entitled "A Multidimensional Scale-Telescoped Filter and Data Extension Package." This method involves successive, linear interpolations in one-, two- and three-dimensions that attempt to construct piece-wise tangents to the good data values. The mean value of the good data within the geometric window is used as a first-guess estimate at all missing data locations before proceeding with an interpolation across missing data regions. The difference between the linearly, interpolated value and the first-guess estimate at the midpoint of a missing data span is used as a correction at the midpoint. Linear interpolation is done again, except between an originally good value and this midpoint, and on both sides of the midpoint. This estimate-and-adjust scheme proceeds, with each successive interpolation being done over half the previous missing data span. This method is analagous to constructing a spline across the missing data span, but is much faster. However, because the first-guess values equal the mean value of the good data, the extrapolation outward from good data is asymptotic to this mean value.
>
> **LINFIT2**— Globally fill all grid points (good and bad) by using a 2-D linear least squares model of the data. First, a 2-D fit is performed at the current level. Then, based on the fit coefficients, all grid points on that level are filled in. This process continues for each level. This method is similar to FILLCON, but is a global fit, not a local.

Decimation can be performed using either of the following two techniques:

> **DECILOC**—Decimate suspicious data based on deviation from local mean. Within the local regions specified by `MAXSTP`, a mean of all good data points is computed. The central (I,J) location is compared to this mean, and if the absolute difference exceeds `MAXDEV`, or there are fewer than `MINPTS` good data points, the central value is flagged as `BAD`. This method is essentially a localized, de-spiking scheme.
>
> **DECIGLO**—Decimate suspicious data based on deviation from the global mean. A standard deviation of good data within each two-dimensional plane is calculated, and all values outside `SIGFAC` standard deviations are flagged as `BAD`. Only those locations within the designated geometric window are affected.
> *Note: This is a single card image command.*

Command structure:

```
PATCHER NAMOUF  NAMINF  METHOD  P5      P6      P7      P8      IFIXAX  IWINDO
PATCHER NAMOUF  NAMINF  FILLCON MAXSTP  MINQAD  MINPTS  DECFLD  IFIXAX  IWINDO
PATCHER NAMOUF  NAMINF  FILLCON3MAXSTP  MINQAD  MINPTS                  IWINDO
```

```
PATCHER NAMOUF  NAMINF  FILLALL NDIM                                    IWINDO
PATCHER NAMOUF  NAMINF  FILLCON3MAXSTP  MINQAD  MINPTS                  IWINDO
PATCHER NAMOUF  NAMINF  LINFIT2 PERC                    DECFLD          IWINDO
PATCHER NAMOUF  NAMINF  DECILOC MAXSTP  MINPTS  MAXDEV          IFIXAX  IWINDO
PATCHER NAMOUF  NAMINF  DECIGLO SIGFAC                                  IWINDO
```

P2	(A)	NAMOUF	Name of the output field containing the results from this command. (DEFAULT = The input field will be updated if NAMOUF is "blank".)
P3	(A)	NAMINF	Name of the input field to be altered.
P4	(A)	METHOD	The method of filling (interpolation/extrapolation) or decimation. Depending on the `METHOD` specified, the remaining information on the card image will be interpreted accordingly:

`---METHOD (P4) set to FILLCON---P3-P10 contain parameterization---`

P5	(F)	MAXSTP	Maximum number of steps (grid intervals) to search outward from a missing data location to determine if it is bounded. (DEFAULT = `5.0`)
P6	(F)	MINQAD	Minimum number of quadrants occupied in order to satisfy the search. (DEFAULT = `3.0`)
P7	(F)	MINPTS	Minimum number of surrounding points required. (DEFAULT = `4.0`)
P8	(F)	DECFLD	Decision field name. Grid location will be filled only if the corresponding grid location in `DECFLD` is also missing.
P9	(A)	IFIXAX	Axis to be fixed for two-dimensional data filling: `Z` z-axis fixed, operate in xy-planes; `Y` y-axis fixed, operate in xz-planes; `X` x-axis fixed, operate in yz-planes. (DEFAULT = `Z`)
P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

`---METHOD (P4) set to FILLCON3---P3-P10 contain parameterization---`

P5	(F)	MAXSTP	Maximum number of steps (grid intervals) to search outward from a missing data location to determine if it is bounded. (DEFAULT = `5.0`)
P6	(F)	MINQAD	Minimum number of octants occupied in order to satisfy the search. (DEFAULT = `3.0`)
P7	(F)	MINPTS	Minimum number of surrounding points required. (DEFAULT = `4.0`)
P8-9		UNUSED	++++++++++++++++++++
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

`---METHOD (P4) set to FILLALL---P3-P10 contain parameterization---`

P5	(F)	NDIM	The number of dimensions for the data filling procedure: `1.0` one-dimensional fill performed line by line in the direction of the axis held fixed; `2.0` two-dimensional fill performed plane by plane along the axis held fixed; `3.0` three-dimensional fill using all the points within the windowed region of the volume. (DEFAULT = `2.0`)
P6-8		UNUSED	+++++++++++++++++++
P9	(A)	IFIXAX	Fixed axis. (See above description.)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

`---METHOD (P4) set to LINFIT2---P3-P10 contain parameterization---`

P5	(F)	PERC	Minimum acceptable percentage of points within window region of plane (DEFAULT = `50.0`)
P6-7		UNUSED	+++++++++++++++++++
P8	(F)	DECFLD	Decision field name. Grid location will be filled only if the corresponding grid location in `DECFLD` is also missing.
P9	(A)	UNUSED	+++++++++++++++++++
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

`---METHOD (P4) set to DECILOC---P3-P10 contain parameterization---`

P5	(F)	MAXSTP	Maximum number fo steps to search outward from each (I,J) grid location. (DEFAULT = `3.0`)
P6	(F)	MINPTS	Minimum number of points required in the region surrounding each (I,J) location. (DEFAULT = `4.0`)
P7	(F)	MAXDEV	Maximum permissible deviation from the local mean for a central value to be retained. (DEFAULT = The Nyquist velocity divided by 2)
P8		UNUSED	+++++++++++++++++++
P9	(A)	IFIXAX	Axis to be fixed for two-dimensional decimation: `Z` z-axis fixed, operate in xy-planes; `Y` y-axis fixed, operate in xz-planes; `X` x-axis fixed, operate in yz-planes. (DEFAULT = `Z`)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

`---METHOD (P4) set to DECIGLO---P3-P10 contain parameterization---`

P5	(F)	SIGFAC	Standard deviation tolerance factor. The standard deviation of the entire field within the designated geometric window is computed, and any value more than `SIGFAC` standard deviations from the mean will be set to the `BAD` data flag. (DEFAULT = `2.0`)
P6-9		UNUSED	++++++++++++++++++++
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

QUIT—Terminates execution of CEDRIC. This *must* be the last command in the current job stack. *Note: This command is itself a single card image command, and it requires no parameters.*

Command structure:

```
QUIT
```

READAIR—Reads in an ASCII dataset containing aircraft position information (and possibly wind information) as a function of time for plotting and gridding (using a closest point scheme). The first column in the dataset must be the time in seconds (as an integer), the second must be the latitude (decimal degrees) of the plane at that time, the third longitude and the fourth, elevation in meters above mean sea level (all as reals). The fifth through the seventh can optionally contain the u,v, and w as measured by the aircraft (as reals). The data in the input file must be uniformly spaced in time. One can also use x and y values (km) instead of latitude and longitude values.

Latitude values must be in the range from -90 to +90. Longitude values can follow the convention where positive is East of the Prime Meridian or the convention where positive is West. The user specifies which convention the longitudes follow. Only one aircraft dataset can be active in Cedric at a time. That is, if a second **READAIR** card is encountered, the data from the first will be overwritten in Cedric. Once the data is read in, it can be used in the **GRAPHICS** command for plotting and overlaying on other plots. U and V vector components (if read in) are assumed to be relative to true North and will be rotated automatically if the existing edit volume has a coordinate orientation where the y-axis is not along true North.
This is a stack command and must be terminated by an **END** card. The maximum number of positions that can be read in is 20000. Thus, for 1 second data, about 5 hours worth of track positions can be read in.

Command structure:

```
READAIR LUN     CDIR    OLAT    OLON    DELTAC  COORD   DTAC
        WINDS   UCOL    VCOL    WCOL
        GRID    NAMU    NAMV    NAMW
        ADVECT  ASPD    ADIR    ATIM
END
```

P2	(F)	LUN	Fortran logical unit number of the INPUT file.
P3	(A)	CDIR	Direction convention of longitudes. `WEST` positive long. is west of Prime Mer. `EAST` positive long. is east of Prime Mer. (DEFAULT = `WEST`)
P4	(F)	OLAT	Latitude of the origin. Used to convert aircraft positions to X and Y distances. (DEFAULT = `Value in ID header`)
P5	(F)	OLON	Longitude of the origin. Used to convert aircraft positions to X and Y distances. (DEFAULT = `Value in ID header`)
P6	(F)	DELTAC	Resolution of data to be used in seconds. For example, the input dataset might contain data every second, while you might only want data for plots every 60 seconds. (DEFAULT = `1.0`)
P7	(A)	COORD	Specifies whether aircraft positions are in lat and long or already in xy space (km). `LAT-LONG` indicates positions are in lat and long `XY` indicates positions are in xy space (km)

			(DEFAULT = `LAT-LONG`)
P8	(F)	DTAC	Time +/- the mean volume time for which data will be read in. Thus, if 3600. is specified, data +/- 1 hour of the mean volume time will be read in for plotting or gridding. The mean volume time is calculated from the volume header information. (DEFAULT = `None. User must specify.`)
P9-10		UNUSED	++++++++++++++++++++

READAIR –STACK of ADDITIONAL OPTIONAL CARD IMAGES–

`---WINDS card---P3-5 contain parameterization---`

P3	(F)	UCOL	Column in IUNIT containing u wind component. Must be between 5 and 7, inclusive. Only two of the three components need to be present in IUNIT in order for vectors to be displayed. Which two depends on which planes of data will be displayed (i.e., x-y, y-z, etc.) (DEFAULT = `None.`)
P4	(F)	VCOL	Column in IUNIT containing v wind component. Must be between 5 and 7, inclusive. (DEFAULT = `None.`)
P5	(F)	WCOL	Column in IUNIT containing w wind component. Must be between 5 and 7, inclusive. (DEFAULT = `None.`)

`---ADVECT card---P3-5 contain parameterization---`

P3	(F)	ASPD	Advection speed in m/s. (DEFAULT = `None.`)
P4	(F)	ADIR	Direction, relative to true North, from which storm is approaching. (DEFAULT = `None.`)
P5	(F)	ATIM	Anchor time to which advection should be done. (DEFAULT = `Mean volume time.`)

`---GRID card---P3-5 contain parameterization---`

P3	(F)	NAMU	Name of output U field, if any, to be gridded. Note that a U field column must be specified on the `WINDS` card. (DEFAULT = `None.`)
P4	(F)	NAMV	Name of output V field, if any, to be gridded. Note that a V field column must be specified on the `WINDS` card. (DEFAULT = `None.`)
P5	(F)	NAMW	Name of output W field, if any, to be gridded. A W field column must be specified on the `WINDS` card. (DEFAULT = `None.`)

READSTA—Reads in an ASCII dataset containing mesonet (or other) station locations for use in plot overlays. The input file must have a specific format, most easily understood by looking at an example (shown below). The positions of the stations are given in either latitude and longitude values (decimal degrees) or in x and y (km). Latitude values must be in the range from -90 to +90. Longitude values can follow the convention where positive is East of the Prime Meridian or the convention where positive is West. The user specifies which convention the longitudes follow.

Only one station dataset can be active in Cedric at a time. That is, if a second **READSTA** card is encountered, the data from the first will be overwritten in Cedric. Once the data is read in, it can be used in the **GRAPHICS** command for plotting and overlaying on other plots. Currently, the station positions are NOT adjusted if a **REMAP** is done to translate or rotate the coordinate system. They will be relative to the grid orientation and layout when the station data was originally read in.

Command structure:

```
READSTA LUN     CDIR     OLAT     OLON     COORD
```

P2	(F)	LUN	Fortran logical unit number of the INPUT file.
P3	(A)	CDIR	Direction convention of longitudes. `WEST` positive long. is west of Prime Mer. `EAST` positive long. is east of Prime Mer. (DEFAULT = `WEST`)
P4	(F)	OLAT	Latitude of the origin. Used to convert station positions to X and Y distances. (DEFAULT = `Value in ID header`)
P5	(F)	OLON	Longitude of the origin. Used to convert aircraft positions to X and Y distances. (DEFAULT = `Value in ID header`)
P6	(A)	COORD	Specifies whether aircraft positions are in lat and long or already in xy space (km). `LAT-LONG`indicates positions are in lat and long `XY`indicates positions are in xy space (km) (DEFAULT = `LAT-LONG`)
P7-10		UNUSED	++++++++++++++++++++

READVOL—Positions a file (that may or may not have multiple volumes) for input to a *current* edit volume. Positioning is done by volume name, time or simply the `NEXT` volume. If a user-defined coordinate system has been established before transfer using the **CREATE** command, the input volume will be mapped into that coordinate system. However, the volume to be transferred **MUST** have grid points in common with the **CREATE**d grid since this transfer involves only movement of fields in INDEX space. This means that the **CREATE**d grid spacings **MUST** be integer multiples of those in the volume to be transferred. If no user-defined coordinate system is active, the edit file coordinate system is defined by the header information of the volume to be transferred.

After this command, the original input file that was accessed is positioned immediately before the next volume in the file, or at the end of the file if no more volumes are present. All searches are forward from the current position within the input file. If the desired volume is "ahead" of the current volume in the input file, the file pointers must be "rewound" to the beginning of the input file before initiating a search. Otherwise, CEDRIC will search forward until the end of the file and issue an error that it could not find the volume.

Whenever a **READVOL** is invoked, the contents of the *current* edit volume in CEDRIC will be completely overwritten. Therefore, the **WRITVOL** command *must* be used to save the *current* edit file that has been operated on, if it is to be preserved. *Note: This is a single card image command.*

Command structure:

```
READVOL LUN     NAMVOL  IBEGTM  IENDTM  IREWND
```

P2	(F)	LUN	Fortran logical unit number of the INPUT file.
P3	(A)	NAMVOL	Volume name. If an actual volume name is specified, a forward search will continue until that volume is located. Otherwise, NAMVOL can simply be `NEXT` to tranfer the next volume in the INPUT file within the time range specified by `IBEGTM` and `IENDTM`. (DEFAULT = `NEXT`)
P4	(F)	IBEGTM	Beginning time (HHMMSS) when NAMVOL is set to `NEXT`. (DEFAULT = `000000.0`)
P5	(F)	IENDTM	Ending time (HHMMSS) when NAMVOL is set to `NEXT`. (DEFAULT = `240000.0`)
P6	(A)	IREWND	Rewind option: `YES` rewind the file pointer to the beginning of the file before initiating the search; `NO` do **NOT** rewind the file pointer before initiating the search. (DEFAULT = `NO`)
P7-10		UNUSED	+++++++++++++++++++

REGRESS—Computes the linear regression statistics between two fields. The linear correlation coefficient (also called the product-moment correlation coefficient, or Pearson's r), standard error of estimate and the equation of the linear least-squares curve fit are produced as a tabular display by plane and/or by volume for all locations with collocated non-missing values within the designated windowed region. *Note: This is a single card image command.*

Command structure:

```
REGRESS IDEST   IFIXAX  SKIP    ITYP    NAMIND  NAMDEP                  IWINDO
```

P2	(A)	IDEST	Destination of the regression output: `PRINT` output will be sent to the printer file; `MICRO` output will be sent to microfiche/microfilm; `BOTH` output will be sent to both printer and film. The only available option is `PRINT`; the others have been documented here for historical reasons. (DEFAULT = `PRINT`)
P3	(A)	IFIXAX	Axis to be fixed for regression analysis: `Z` z-axis fixed, regression is done in xy-planes; `Y` y-axis fixed, regression is done in xz-planes; `X` x-axis fixed, regression is done in yz-planes. (DEFAULT = `Z`)
P4	(F)	SKIP	Increment between output levels. A regression analysis and output will be generated every SKIP levels. (DEFAULT = `1.0`)
P5	(A)	ITYP	Type of regression analysis to be done: `PLANE` regression analysis will be done by two-dimensional planes; `VOLUME` regression analysis will be done by volume, (all points within the windowed region); `BOTH` regression analysis will be done both by plane and by volume. (DEFAULT = `BOTH`)
P6	(A)	NAMIND	Field name of the independent variable to be used in the regression analysis. (DEFAULT = This field name *must* be specified.)
P7	(A)	NAMDEP	Field name of the dependent variable to be used in the regression analysis. (DEFAULT = This field name *must* be specified.)
P8-9		UNUSED	++++++++++++++++++++
P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

RELAXUV—Computes a new set of horizontal wind component fields (UNEW,VNEW) using the current fields (UCUR,VCUR) and the vertical air motion field (WAIR) derived from a variationally adjusted integration. That integration scheme does not adjust (redistribute random errors vertically) the horizontal winds, only the horizontal convergence so there is no guarantee that three-dimensional convergence is exactly zero as required in the form of the mass continuity equation outlined in **Appendix F**. Therefore, the (UCUR,VCUR) components are iteratively corrected so as to minimize the mean difference between horizontal and vertical convergence within each level. The same density profile that was used in the integration should be employed here as well. The user may want to generate a new horizontal convergence field using (UNEW,VNEW) once this command has been executed. *Note: This is a single card image command.*

Command structure:

```
RELAXUV UNEW     VNEW     UCUR     VCUR     WAIR     DWFAC    CNCRIT  ZLOWER  ZUPPER
```

P2	(A)	UNEW	New U-component output field name.
P3	(A)	VNEW	New V-component output field name.
P4	(A)	UCUR	Existing U-component input field name.
P5	(A)	VCUR	Existing V-component input field name.
P6	(A)	WAIR	Variationally adjusted W-component input field name.
P7	(H)	DWFAC	Density weighting factor. This parameter is in the form `Z*nnnnnn` where `Z*` is typed literally and `nnnnnn` is the user supplied term in the density weighting function: `RHO = EXP ( - Z * nnnnnn ),` where Z is the height in kilometers (MSL) at each level. If `DWFAC` is not blank and the preamble `Z*` is missing, density weighting will **NOT** be used. Also, any sequence of characters such as `UNITY` or `NODENS` will turn off density weighting. (DEFAULT = `Z*0.1`)
P8	(H)	CNCRIT	Criteria for determining procedure convergence at each level. This parameter is in the form `III.EEEE`, where `III` is the maximum number of iterations to perform at each level while attempting to satisfy the requirement that the mean difference between the vertical and horizontal convergence at each level be less than `EEEE`. The part `III` must be an integer whose absolute value is in the range `1-25`. If `III` is negative, CEDRIC will iterate exactly `ABS(III)` times, regardless of other convergence criteria. The part `EEEE` must be in the range `0.01-0.99`. (DEFAULT = `10.01`); however each part is DEFAULTed separately.
P9	(H)	ZLOWER	Lowest Z-level to be included in the procedure: `I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level; `D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level. (DEFAULT=lowest level in the volume).
P10	(H)	ZUPPER	Highest Z-level to be included in the procedure: `I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level; `D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level. (DEFAULT=highest level in the volume).

REMAP—Remaps the data from all existing fields to a new user-specified Cartesian coordinate system. This command is useful for adjusting individual volumes to a storm-relative coordinate system (translation) and for generating vertical sections along the direction of storm motion (rotation). This command also allows the user to interpolate a volume from a **coplane or elevation** coordinate system to a three-dimensional **Cartesian** coordinate system. The total number of data points, grid points times the number of fields—Nx * Ny * Nz * Nf—has the same restrictions as those imposed on all CEDRIC files. There are no other restrictions in the specification of the new coordinate system. Its origin can be translated, the X and Y axes rotated about Z, and the grid spacing and layout itself can be changed. If only a shifting of the data has been requested, i.e. a translation of the origin by an integral number of grid spacings in each orthogonal direction, no interpolation will be performed, only shifting. Otherwise, a bilinear/closest point scheme of three-dimensional interpolation will be done. If the **REMAP** Z-levels are the same as those of the *current* edit volume to be remapped, a two-dimensional interpolation is done only in xy-planes. This is a stack command and *must* be terminated with an **END** command.

The additional stack card images are used to specify the grid and any special processing. If the seventh parameter (P7) of the **REMAP** command is `NEWGRID`, the next card image *must* specify the grid. If P7 is not set to `NEWGRID`, then subsequent card images specify any special processing. This includes interpolation of vector components, especially when the coordinate system is rotated. In that case, the vector is properly transformed and its components interpolated. When radar radial velocities are interpolated, a local unfolding scheme can be invoked in order to preserve the discontinuities associated with folded velocities. See Miller et al. (1986) for the details of this procedure. Additionally, fields that are represented on a dB scale can be converted to a linear scale, interpolated, and then converted back to a dB scale.

Command structure:

```
REMAP   ICORD   ANGXAX  XORGIN  YORGIN  ZORGIN  NWGRD   MAXREL
        X1      X2      XD      Y1      Y2      YD      Z1      Z2      ZD
        SPEC    P3      P4      P5      P6      P7      P8      P9      P10
        ROTATE  INX     INY     INZ     OUTX    OUTY    OUTZ
        LINEAR  FIELD
        UNFOLD  FIELD   NYQ
END
```

P2	(A)	ICORD	Coordinate system of data before interpolation or shifting: `CART` interpolate from a Cartesian coordinate system; `COPLANE` interpolate from a coplane coordinate system; `ELEV` interpolate from a constant elevation angle coordinate system. These are currently the only possibilities. The coplane angles are assumed to be equally-spaced, and the elevation angles are assumed to be unequally-spaced. Actual elevation angles are taken from information in the level header rather than the equally-spaced coordinates of the 510-word CEDRIC volume header. If `ICORD` is left blank, CEDRIC determines the input coordinate system.
P3	(F)	ANGXAX	Horizontal azimuth angle of the new X-axis in the new coordinate system, measured clockwise from true north. If `ANGXAX` is specified and is **NOT** equal to the existing angle of the X-axis, the new coordinate system will be defined, assuming rotation about the vertical or Z-axis. (DEFAULT = same as that of the old coordinate system.)
P4	(F)	XORGIN	The X-coordinate in the **OLD** space where the origin of the **NEW** coordinate system is to be located. (DEFAULT = `0.0`), there is no change.

P5	(F)	YORGIN	The Y-coordinate in the **OLD** space where the origin of the **NEW** coordinate system is to be located. (DEFAULT = `0.0`), there is no change.
P6	(F)	ZORGIN	The Z-coordinate in the **OLD** space where the origin of the **NEW** coordinate system is to be located. (DEFAULT = `0.0`), there is no change.
P7	(A)	NWGRD	Flag that indicates whether a new grid specification card will follow this one: `NEWGRID` the next card specifies the new grid layout; the old grid will be used. (DEFAULT =); "blank", no new grid will be specified.
P8	(F)	MAXREL	Maximum distance (kilometers) to relocate a point when closest point method of interpolation is being used. (DEFAULT = `0.0`)
P9-10		UNUSED	+++++++++++++++++++

—NWGRD (P7) set to NEWGRID—NEXT CARD IMAGE MUST SPECIFY GRID—

P2	(F)	X1	Beginning X-coordinate of the new grid. There is no DEFAULT; the user must specify.
P3	(F)	X2	Ending X-coordinate of the new grid. There is no DEFAULT; the user must specify.
P4	(F)	XD	Spacing along X-axis. There is no DEFAULT; the user must specify.
P5	(F)	Y1	Beginning Y-coordinate of the new grid.There is no DEFAULT; the user must specify.
P6	(F)	Y2	Ending Y-coordinate. There is no DEFAULT; the user must specify.
P7	(F)	YD	Spacing along Y-axis. There is no DEFAULT; the user must specify.
P8	(F)	Z1	Beginning Z-coordinate of the new grid.There is no DEFAULT; the user must specify.
P9	(F)	Z2	Ending Z-coordinate. There is no DEFAULT; the user must specify.
P10	(F)	ZD	Spacing along Z-axis. There is no DEFAULT; the user must specify.

REMAP –STACK of ADDITIONAL CARD IMAGES–

P2 .	(H)	SPEC	Option for special processing of any fields during remapping `ROTATE` a three-dimensional vector will be interpolated. No interpolation of vector components is allowed for constant elevation to cartesian. `UNFOLD` local unfolding will be done during interpolation; `LINEAR` the field will be converted from a dB scale before interpolation and back to a dB scale after interpolation. There is no DEFAULT; the user must specify.

```
---SPEC (P2) set to ROTATE---P3-8 contain parameterization---
```

P3	(A)	INX	Name of input vector component field that corresponds to motion in the **OLD** X-direction. This name must be an existing field. There is no DEFAULT; the user must specify.
P4	(A)	INY	Name of input vector component field that corresponds to motion in the **OLD** Y-direction. This name must be an existing field. There is no DEFAULT; the user must specify.
P5	(A)	INZ	Name of input vector component field that corresponds to motion in the **OLD** Z-direction. This name must be an existing field. There is no DEFAULT; the user must specify.
P6	(A)	OUTX	Name of output vector component field that will correspond to motion in the **NEW** X-direction. This name can refer to an existing field (including any of P3-P5), or if it does not exist, the field will be created. The user should be certain that there is enough memory (Nx * Ny * Nz * Nf) to create any new fields. If a new field is implied and not enough memory exists, the program will print out an error message indicating insufficient memory and exit. There is no DEFAULT; the user must specify.
P7	(A)	OUTY	Name of output vector component field that will correspond to motion in the **NEW** Y-direction. This name can refer to an existing field (including any of P3-P5), or if it does not exist, the field will be created. See the number of grid points times number of fields issue described above. There is no DEFAULT; the user must specify.
P8	(A)	OUTZ	Name of output vector component field that will correspond to motion in the **NEW** Z-direction. This name can refer to an existing field (including any of P3-P5), or if it does not exist, the field will be created. See the number of grid points times number of fields issue described above. There is no DEFAULT; the user must specify.
P9-10		UNUSED	+++++++++++++++++++

```
---SPEC (P2) set to LINEAR---P3 contains parameterization---
```

An outline of the linearization procedure is:

```
        r = original dB value
        r = 10**(r/10.0)
   interpolation of points is done
     fval = final value from interpolation
     fval = 10*log10(fval)
```

P3	(A)	FIELD	Name of field to be made linear for interpolation. This name *must* refer to an existing field name. There is no DEFAULT; the user must specify.
P4-10		UNUSED	+++++++++++++++++++

```
---SPEC (P2) set to UNFOLD---P3-4 contain parameterization---
```

Radial velocity fields are to be locally unfolded.

P3	(A)	FIELD	Name of the radial velocity field. There is no DEFAULT; the user must specify.
P4	(F)	NYQ	Nyquist velocity to be used for unfolding. (DEFAULT=Use value in housekeeping.)

P5-10	UNUSED	++++++++++++++++++++

END This **END** command terminates the **REMAP** command stack.

RENAME—Allows the user to rename existing fields in the current edit volume. The previous name is replaced by a new one and all references to the old name will no longer be valid. *Note: This is a single card image command.*

Command structure:

```
RENAME  NAMOLD  TO      NAMNEW  NAMOLD  TO      NAMNEW  NAMOLD  TO      NAMNEW
```

P2	(A)	NAMOLD	Old name of field to be renamed.
P3		UNUSED	++++++++++++++++++++The addition of `TO` improves readability.
P4	(A)	NAMNEW	New name of field specified by P2.
P5	(A)	NAMOLD	Old name of field to be renamed.
P6		UNUSED	++++++++++++++++++++
P7	(A)	NAMNEW	New name of field specified by P5.
P8	(A)	NAMOLD	Old name of field to be renamed.
P9		UNUSED	++++++++++++++++++++
P10	(A)	NAMNEW	New name of field specified by P7.

SAMPLER—Computes a radial velocity field for a fixed radar position (X,Y,Z) using the orthogonal components (U,V,W) of a three-dimensional wind field. The radar position and the vector components are assumed to be relative to the same set of coordinate axes. The user can specify that the radar coordinates supplied and the vector components are relative to the coordinate system where the y-axis points along true north, or that the radar coordinates and the vector components are relative to the current coordinate system, whatever its orientation. However, in both cases the radar coordinates supplied are relative to the current origin, only the orientation of the axes can be chosen.
Note: This is a single card image command.

Command structure:

```
SAMPLER NAMOUF  NAMU    NAMV    NAMW    XRAD    YRAD    ZRAD    COORD   IWINDO
```

P2	(A)	NAMOUF	Name of the radial velocity output field.
P3	(A)	NAMU	Name of the U-component wind field. (DEFAULT=The U-component field will be set to `0.0`)
P4	(A)	NAMV	Name of the V-component wind field. (DEFAULT=The V-component field will be set to `0.0`)
P5	(A)	NAMW	Name of the W-component wind field. If a W-component field is specified any missing data values contained within it will be replaced by `W=0.0` when computing radial velocity. (DEFAULT=The W-component field will be set to `0.0`)
P6	(F)	XRAD	X coordinate of the sampling position. (DEFAULT = `0.0`)
P7	(F)	YRAD	Y coordinate of the sampling position. (DEFAULT = `0.0`)
P8	(F)	ZRAD	Z coordinate of the sampling position. (DEFAULT = `0.0`)
P9	(A)	COORD	Coordinate axes orientation. `NORTH` radar coordinates and wind vector components assumed to be relative to the coordinate system where the y-axis points towards true north. `REL` radar coordinates and wind vector components assumed to be relative to the *current* coordinate system orientation, whatever it may be. (DEFAULT = `REL`)
P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

SHIFTER—Advection of data at horizontal levels according to analysis time or time to which data is shifted, storm motion, and data or collection time. Two techniques are available:

> **UNIFORM**—shift the data within each level an integral number of grid points in X and Y. This number of grid positions is calculated based on the average collection time of the individual Z-levels, unless otherwise specified.
>
> **DIFFERENTIAL**—relocates data values within a level along X and Y according to individual grid point collection times. The advected field is subsequently remapped to the regular (X,Y) analysis grid using bi-linear interpolation.

If a time field does not exist and a fixed time would not be representative for every (X,Y,Z) location in the volume, a simple **FUNCTION** stack can be devised by the user to generate an appropriate time field. Time fields are always defined as the time in seconds corresponding to each (X,Y,Z) location after the time of day (HHMMSS) associated with the beginning of the volume scan. The **SHIFTER** command should be avoided unless one is working with information from a single radar. When combining multiple radars, the advection feature in the **SYNTHES** command instead should be invoked since it will produce the same result without the necessity of advecting each radar volume explicitly using the **SHIFTER** command. It is generally unwise to advect individual fields, rather all fields should be advected when using this command. Once advection is completed, this command replaces the beginning and ending times in the edit volume header (words 116–127) with the analysis time (ITANAL). It will be necessary to use the **FIXIDS** command should you wish to shift individual levels within a volume so that the original times continue to be used after the first set of levels are advected. *Note: This is a single card image command.*

Command structure:

```
SHIFTER NAMOUF  NAMINF  NAMTIM  ITANAL  STMDIR  STMSPD  IADTYP  ZLOWER  ZUPPER
```

P2 (A)	NAMOUF	Name of the advected output field. If `ALL` is specified, all fields will be advected. (DEFAULT=The input field will be updated.)
P3 (A)	NAMINF	Name of the input field to be advected. If `ALL` is specified, all fields will be advected. (DEFAULT=If neither NAMOUF or NAMINF are specified, all fields will be advected.)
P4 (A)	NAMTIM	Time field specifier. This may be either an actual edit field name or a special mnemonic indicating that the TIME field is to be set to a constant. This special mnemonic must be in the following format: `T=nnnnnn` where `nnnnnn` is a 6 digit floating point value specifying the constant to which the TIME field will be set. (DEFAULT = `T=0.0`)
P5 (F)	ITANAL	Analysis time to be used as a reference for advecting the data in `HHMMSS` (time of day.) (DEFAULT=Starting time of the *current* edit volume.)
P6 (F)	STMDIR	Storm direction. This is defined as degrees clockwise from North in the direction from which the storm is approaching. (Same as a wind is defined.) (DEFAULT = `0.0`)
P7 (F)	STMSPD	Storm speed (M/S). (DEFAULT = `0.0`), no advection will be performed.)
P8 (A)	IADTYP	Method of advection: `UNI` uniform advection will be performed; `DIF` differential advection will be performed. (DEFAULT = `DIF`)

P9 (H) ZLOWER — Lowest Z-level to be included in the advection procedure:
`I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level;
`D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level;
(DEFAULT=lowest level in the volume).

P10 (H) ZUPPER — Highest Z-level to be included in the advection procedure:
`I=nnnnnn` level is specified in INDEX space and `nnnnnn` is the index of the desired level;
`D=nnnnnn` level is specified in DISTANCE space and `nnnnnn` is the height (KM) of the desired level;
(DEFAULT=highest level in the volume).

STATS—Generates a statistical display for a set of designated fields. The display includes mean, standard deviation, minimum-maximum values, and number of good data values. *Note: This is a single card image command.*

Command structure:

```
STATS   IDEST   IFIXAX  SKIP   NMSPEC  ITYP                              IWINDO
```

P2	(A)	IDEST	Destination of the display. The only available option is `PRINT`.
P3	(A)	IFIXAX	Axis to be fixed for displays: `Z` z-axis fixed, display xy-planes; `Y` y-axis fixed, display xz-planes; `X` x-axis fixed, display yz-planes. (DEFAULT = `Z`)
P4	(F)	SKIP	Increment between display levels. A display will be generated every SKIP levels. (DEFAULT = `1.0`)
P5	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` only the primary field group will be displayed; `SEC` only the secondary field group will be displayed; `ALL` every field will be displayed. See the **FIELDSET** command for instructions on how to set primary and secondary field groups. (DEFAULT = `PRI`)
P6	(A)	ITYP	Region for computing the statistics: `PLANE` statistics will be generated for each plane; `VOLUME` statistics will be generated for the volume; `BOTH` statistics will be generated for each plane and for the volume. (DEFAULT = `BOTH`)
P7-9		UNUSED	++++++++++++++++++++
P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

STPLOT—Generates plotted profiles of selected statistics along any fixed geometric axis for a set of designated fields. Profiles for the following five quantities can be produced: Minimum, Mean - scale*σ, Mean value, Mean + scale*σ, and Maximum. The quantity σ is the standard deviation. Each of these statistics is determined within two-dimensional planes, according to which axis is fixed and over the windowed region. Statistics are then plotted as a function of the fixed axis direction. The specified window determines the limits of the geometric axis of the profile, and either the user or the program decides the ranges for the statistics axis. Profiles may be oriented in either the vertical or horizontal direction. This is a stack operation and *must* be terminated by an **END** command. The stack consists of additional card images that allow the user to specify the fields to be examined, along with associated display parameterizations. If a field is referred to more than once, the last specification will be used to display the field. Fields will be displayed according to their order within the CEDRIC program and not according to their order in the stack.

Command structure:

```
STPLOT  IDEST   IFIXAX  IORENT  MEAN    SIGMA   MIN     MAX     SDFAC   IWINDO
        NMSPEC  STMIN   STMAX
END
```

P2 (A) IDEST — Destination of the display. The only available option is `MICRO`.

P3 (A) IFIXAX — Axis to be fixed for displays:
- `Z` z-axis fixed, display xy-planes;
- `Y` y-axis fixed, display xz-planes;
- `X` x-axis fixed, display yz-planes.

(DEFAULT = `Z`)

P4 (A) IORENT — Orientation of the statistical profiles:
- `HOR` statistical profiles will be plotted left-to-right;
- `VER` statistical profiles will be plotted bottom-to-top.

(DEFAULT = `VER`)

P5 (H) MEAN — Specification for MEAN profile. Profile specifications are made in the following 8-character format: `P/C/nnnn` where `P`, `C` and `nnnn` are supplied by the user.
- `P` the type of plotted line for the profile:
  - `S` the connecting line is solid;
  - `D` the connecting line is long-dashed;
  - `L` the connecting line is short-dashed;

(DEFAULT=no connecting line will be drawn.)
- `C` a keyboard character to digitize at the location of each datum;

(DEFAULT=no character will be digitized.)
- `nnnn` a 4 digit floating point number designating the size of the character `C` in plotter units (1024 x 1024 units constitutes a frame).

(DEFAULT = `10.0`)

(DEFAULT = MEAN profile will NOT be plotted.)

P6 (H) SIGMA — Specification for the MEAN ± (SIGMA * SDFAC) profiles. Profile specifications are made as described above for the MEAN (P5).

(DEFAULT = SIGMA profiles will NOT be plotted.)

P7 (H) MIN — Specification for the MINimum value profile. Profile specifications are made as described above for the MEAN (P5).

(DEFAULT = MIN profile will NOT be plotted.)

P8	(H)	MAX	Specification for the MAXimum value profile. Profile specifications are made as described above for the MEAN (P5). (DEFAULT = MAX profile will NOT be plotted.) IF no profiles are activated by user-specified parameters P5-8, the program will generate displays containing MEAN and SIGMA profiles using internally determined specifications.
P9	(F)	SDFAC	Sigma scaling factor. SIGMA profiles will be drawn `SDFAC` standard deviations on either side of the MEAN. (DEFAULT = `1.0`)
P10	(A)	IWINDO	Windowing specification for this command: `WINDOW` execute the command only in the current user-set window; `FULL` execute the command over the full (X,Y,Z) grid. (DEFAULT = `FULL`)

STPLOT –STACK of ADDITIONAL CARD IMAGES–

P2	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` only the primary field group will be displayed; `SEC` only the secondary field group will be displayed; `ALL` every field will be displayed. See the **FIELDSET** command for instructions on how to set primary and secondary field groups. (DEFAULT = `PRI`)
P3	(F)	STMIN	Minimum value to display on the profile plot. (DEFAULT = `0.0`)
P4	(F)	STMAX	Maximum value to display on the profile plot. (DEFAULT = `0.0`) If `STMIN = STMAX`, the program decides what the endpoints should be for the profiles specified. Hence, leaving STMIN and STMAX will make the program decide plot limits.
P5-10		UNUSED	++++++++++++++++++++

END This **END** command terminates the **STPLOT** command stack.

SURFACE—Generates a three-dimensional perspective display of a field for each (x,y) from any viewing angle for a set of designated fields. A figure whose height or displacementabove the perspective horizontal plane is proportional to the values in the field will be produced for every constant Z-plane within the display window. This is a stack operation and *must* be terminated by an **END** command. The stack consists of additional card images that allow the user to specify the fields to be examined, along with associated display parameterizations. Only one specification per field is permitted in the stack. If a field is referred to more than once, the last specification will be used to display that field. Fields will be displayed according to their order within the CEDRIC program and not according to their order in the stack.

Command structure:

```
SURFACE IDEST   SKIP    ICONAX  XEYMUL  YEYMUL  FEYMUL  LABTYP          IWINDO
        NMSPEC  VALMIN  VALMAX  VALZER
END
```

P2 (A) IDEST — Destination of the display: The only available option is `MICRO`.

P3 (F) SKIP — Increment between display levels. A display will be generated every SKIP levels.
(DEFAULT = `1.0`)

P4 (A) ICONAX — Axes to be contoured: `X`, `Y` and/or `Z` may appear in any combination in the first three left-justified positions of ICONAX. Any axis that is specified will be contoured.
(DEFAULT = `XY`)

P5 (F) XEYMUL — Multiple of the X-axis length from Xmin where the eye is to be positioned.
(DEFAULT = `2.0`)

P6 (F) YEYMUL — Multiple of the Y-axis length from Ymin where the eye is to be positioned.
(DEFAULT = `-6.0`)

P7 (F) FEYMUL — Multiple of the F-axis length from Fmin where the eye is to be positioned.
(DEFAULT = `2.5`)

P8 (A) LABTYP — Type of axes labeling desired:
- `FULL` the full (X,Y,FIELD) axes legend will be generated;
- `FIELD` only FIELD axis will be labeled;
- `NONE` no axes will be labeled.

(DEFAULT = `FULL`)

P9 UNUSED ++++++++++++++++++++

P10 (A) IWINDO — Windowing specification for this command:
- `WINDOW` execute the command only in the current user-set window;
- `FULL` execute the command over the full (X,Y,Z) grid.

(DEFAULT = `FULL`)

P2	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` only the primary field group will be displayed; `SEC` only the secondary field group will be displayed; `ALL` every field will be displayed. See the **FIELDSET** command for instructions on how to set primary and secondary field groups. (DEFAULT = `PRI`)
P3	(F)	VALMIN	Minimum field value to be depicted in the SURFACE display. Any number less than or equal to this value will appear at the "lowest" point in the display. (DEFAULT = `0.0`)
P4	(F)	VALMAX	Maximum field value to be depicted in the SURFACE display. Any number greater than or equal to this value will appear at the "highest'"point in the display. VALMAX must be greater than VALMIN. (DEFAULT = `50.0`)
P5	(F)	VALZER	Value at ZERO height. FIELD values above VALZER will be depicted as "mountains"; FIELD values below VALZER will be depicted as "valleys". Any missing data values in the field to be displayed will be set to VALZER. VALZER must NOT be less than VALMIN nor greater than VALMAX. (DEFAULT = `0.0`)
P6-10		UNUSED	++++++++++++++++++++

END This **END** command terminates the **SURFACE** command stack.

SYNTHES—Synthesizes radial velocity information from two or more Doppler radars. Each radar is assigned a weight of unity and the least-squares solution described in **Appendix F** is employed for the derivation of the orthogonal components of particle motion. Up to five additional fields may be transferred from each of the input radar volumes. Only those original radar fields that are explicitly transferred by the user will be copied over to the output edit volume produced. Whenever the **SYNTHES** command is invoked the contents of the *current* edit volume in CEDRIC will be completely overwritten. Therefore, the **WRITVOL** command should be used to save the current edit file if it is to be preserved. Any input single-radar files referenced by **SYNTHES** must be predefined in the script being run on the system. This command creates a new volume containing the synthesized wind information, along with any additional fields that the user has designated for transfer from the radar input volumes.

If a user-defined coordinate system has been established using the **CREATE** command, all radar input volumes will be remapped accordingly. If no user-defined coordinate system is active, the grid specifications of the first radar input volume will be used, and all subsequent input volumes will be remapped onto that grid. All radar input volumes **MUST** have grid points in common; that is, grid spacings can be only integer multiples of each other, including the spacings of any **CREATE**d grids. The synthesis can be done in a three-dimensional Cartesian coordinate systems or in a coplane coordinate systems. By default, CEDRIC determines the coordinate system from the volume header. The volume header information can be overridden with the **COORD** command.

SYNTHES produces the following fields (see **Appendix F** for further details):

`U` the U-component of particle motion.
`V` the V-component of particle motion.
`W` the W-component of particle motion.
`CT` a bit map of the radars contributing to the synthesis at each (X,Y,Z) location. The rightmost bit is set to one (1) if the first radar was used, 2nd bit for the 2nd radar and so forth. The actual number of contributing radars can be calculated using the **BITCOUNT** primitive in the **FUNCTION** command.
`USTD` the normalized standard deviation of the U-component.
`VSTD` the normalized standard deviation of the V-component.
`WSTD` the normalized standard deviation of the W-component. Whenever the two-equation solution option is invoked or when only two radars are used as input, `W` and `WSTD` cannot be computed. Instead, the following two fields are generated:
`EWU` the multiplier of (an unknown) `W` to adjust the two-equation `U` to obtain a “true” `U`.
`EWV` the multiplier of (an unknown) `W` to adjust the two-equation `V` to obtain a “true” `V`.
Whenever an overdetermined solution is applied, that is, the number of radars exceeds the number of unknowns `U,V,W`, an additional field is generated:
`MPE` the “Most Probable Error” field defined as follows:

$$MPE = \frac{\Sigma\left[V_n - V_{rn}^2(U,V,W)\right]^{1/2}}{(N-M)}$$

where V_n is the nth input radial velocity at synthesis grid point, V_{rn} is the resampled radial velocity from the synthesized winds, `N` is the number of radars, and `M(=2 or 3)` is the number of synthesized wind components (2 or 3). `W` is assumed to be zero when a two-equation or (U,V) solution is specified by the user.

Once the **SYNTHES** command has finished execution, its output volume is available for editing, or any other manipulation and preservation in the same manner as any edit volume. This command is a stack operation and *must* be terminated by an **END** command. The stack consists of additional card images that contain parameterization information associated with each of the radar input volumes. The parameterization includes the volume specification as well as the velocity field name and advection information. If any input volume is from an **AIRBORNE** Doppler radar (Landmark #2 = `AIRBRN`), azimuth and elevation angle field names must be supplied in place of any advection information. See the descriptions of P8 and P9 in the stack for further details. Also included on **EXTRA** card images are the names of any additional fields to be transferred from the individual radars.

Command structure:

```
SYNTHES NSNVOL  NRADS   DTEST1  DTEST2  DTEST3  ITWOEQ  ITANAL  STMDIR  STMSPD
        LUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMVEL  LADTYP  NAMTIM  NADFSW
        INPUT   NAMINP  NAMINP  NAMINP  NAMINP  NAMINP
        OUTPUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT
        LUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMVEL  LADTYP  NAMTIM  NADFSW
        INPUT   NAMINP  NAMINP  NAMINP  NAMINP  NAMINP
        OUTPUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT
        LUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMVEL  LADTYP  NAMTIM  NADFSW
        INPUT   NAMINP  NAMINP  NAMINP  NAMINP  NAMINP
        OUTPUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT
END
```

P2 (A) NSNVOL — Name of the SYNTHES output volume.
(DEFAULT=Name of the first radar input volume.)

P3 (F) NRADS — Number of radar input volumes to be included in the synthesis. A card image containing relevant parameterization information must be supplied in the **SYNTHES** stack for each radar input volume. From two to six input radars may be specified.

P4 (F) DTEST1 — User-specified constraint on the potential contribution of `W` to the `(U,V)` solutions. Both `EWU` and `EWV` must be less than DTEST1 for an acceptable two-equation solution. See **Appendix F** for details. (DEFAULT = `1.0`)

P5 (F) DTEST2 — User-specified constraint on the quality of the `(U,V)` solution. Both `USTD` and `VSTD` must be less than DTEST2 for an acceptable two-equation solution. See **Appendix F** for details. (DEFAULT = `100.0`)

P6 (F) DTEST3 — User-specified constraint on the quality of the `W` solution. `WSTD` must be less than DTEST3 for an acceptable three-equation solution for `U, V` and `W`. See **Appendix F** for details. (DEFAULT = `100.0`)

P7 (A) ITWOEQ — Two-equation solution `(U,V)` option:
`YES` use the two-equation approximation as described in **Appendix F** at every grid location and set `W` to the missing data value;
`NO` use the two-equation approximation at those locations with data only from two radars; otherwise, attempt a three-equation solution.
(DEFAULT = `NO`)

P8 (F) ITANAL — Analysis time to be used as a reference for advecting the data in `HHMMSS` (time of day.) If no advection is to be performed, P8 thru P10 should be left "blank".
(DEFAULT=Starting time of the first radar input volume.)

P9 (F) STMDIR — Storm direction for advecting data. This is defined as degrees clockwise from North in the direction from which the storm is approaching. (Same as a wind is defined.)
(DEFAULT = `0.0`)

P10 (F) STMSPD — Storm speed (M/S).
(DEFAULT = `0.0`, no advection will be performed.)

P2 (F) LUNIT	Fortran logical unit number of the radar input volume. LUNIT must be predefined in the Job Control Language.
P3 (A) NAMVOL	Volume name: If an actual volume name is specified, a forward search will continue until that volume is located. No additional constraints are imposed on the search; If NAMVOL = `NEXT`, the next volume of the radar file within the time range specified by `IBEGTM` and `IENDTM` will be transferred into the synthesis procedure. (DEFAULT = `NEXT`)
P4 (F) IBEGTM	Beginning time (HHMMSS) when NAMVOL is set to `NEXT`. (DEFAULT = `000000.0`)
P5 (F) IENDTM	Ending time (HHMMSS) when NAMVOL is set to `NEXT`. (DEFAULT = `240000.0`)
P6 (A) IREWND	Rewind option: `YES` rewind the file pointers before initiating the search; `NO` do NOT rewind the file pointers before initiating the search. (DEFAULT = `NO`)
P7 (A) NAMVEL	Name of the radial velocity field from this radar to be used in the synthesis. This field must be specified.
P8 (A) IADTYP	If the input volume is from a ground-based Doppler radar `IADTYP` is the advection method to employ: `UNI` uniform advection will be performed; `DIF` differential advection will be performed; `NONE` no advection will be performed. (DEFAULT = `NONE`) If landmark #2 is set to `AIRBRN`, the input volume is assumed to be from an airborne Doppler radar and `IADTYP` must contain the name of the `AZIMUTH` field.
P9 (A) NAMTIM	If the input volume is from a ground-based Doppler radar and advection is to be performed, `NAMTIM` is the time field specifier. This may be either an actual edit field name or a special mnemonic indicating that the time field is to be set to a constant. This special mnemonic must be in the following format: `T=nnnnnn` where `nnnnnn` is a 6 digit floating point value specifying the constant to which the time field will be set. (DEFAULT = `T=0.0`) If the input volume is from an airborne Doppler `NAMTIM` must contain the name of the `ELEVATION` field.

P10	(A)	NADFSW	Additional field switch: `YES` additional fields will be transferred from this radar and added to the **SYNTHES** output volume; `NO` no fields will be transferred from this radar to the output volume. (DEFAULT = `NO`)

SYNTHES –STACK of EXTRA CARD IMAGES–

Whenever NADFSW is set to `YES`, two (2) **EXTRA** card images *must* immediately follow. The first card image `(INPUT)` contains the names of the fields to be transferred as they appear in the radar INPUT volume. The second card image `(OUTPUT)` contains the corresponding field names as they are to be designated in the output volume. This is necessary to avoid any duplicate names in the **SYNTHES** output volume. The keyword `INPUT` *must* be present in parameter position P2 of the first **EXTRA** card image and `OUTPUT` must be present in position P2 of the second card image.

P2	(A)	INPUT	`INPUT` must be present and the remaining parameters contain the names of fields to be input from this radar.
P3-7	(A)	NAMINP	Names of the fields to be transferred from the radar input volume. As many as five (5) fields may be specified from each radar.
P8-10		UNUSED	++++++++++++++++++++
P2	(A)	OUTPUT	`OUTPUT` must be present and the remaining parameters contain the names of fields in the output **SYNTHES** volume.
P3-7	(A)	NAMOUT	Destination names of the fields to be transferred from the radar input volume. As many as five (5) fields may be specified from each radar.
P8-10		UNUSED	++++++++++++++++++++

END This **END** command terminates the **SYNTHES** command stack.

THREED—Generates a three-dimensional isosurface display from any viewing angle for a set of designated fields. This is a stack operation and *must* be terminated by an **END** command. The stack consists of additional card images that allow the user to specify the fields to be examined, along with associated display parameterizations. If a field is referred to more than once, the last specification will be used to display the field. Fields will be displayed according to their order within the CEDRIC program and not according to their order in the stack.

Command structure:

```
THREED  IDEST   ICONAX  XSTR    YSTR    ZSTR    XEYE    YEYE    ZEYE
        NMSPEC  CLEVEL  IVISBL
END
```

P2	(A)	IDEST	Destination of the display. The only available option is `MICRO`.
P4	(A)	ICONAX	Axes to be contoured: `X, Y` and `Z` may appear in any combination in the first three left-justified positions of ICONAX. Any axis that is specified will be contoured. (DEFAULT = `XY`)
P4	(F)	XSTR	X-axis stretching factor. (DEFAULT = `1.0`)
P5	(F)	YSTR	Y-axis stretching factor. (DEFAULT = `1.0`)
P6	(F)	ZSTR	Z-axis stretching factor. (DEFAULT = `1.0`)
P7	(F)	XEYE	X-coordinate of the eye position. (DEFAULT= `5.0` times the length of the X-axis length from the Xmin location.
P8	(F)	YEYE	Y-coordinate of the eye position. (DEFAULT= `-3.5` times the length of the Y-axis length from the Ymin location.
P9	(F)	ZEYE	Z-coordinate of the eye position. (DEFAULT= `2.0` times the length of the Z-axis length from the Zmin location.
P10		UNUSED	++++++++++++++++++++

THREED –STACK of ADDITIONAL CARD IMAGES–

P2	(A)	NMSPEC	Field name specifier. This may be either an actual edit field name or one of the following codes designating a predetermined group of fields: `PRI` only the primary field group will be displayed; `SEC` only the secondary field group will be displayed; `ALL` every field will be displayed. See the **FIELDSET** command for instructions on how to set primary and secondary field groups. (DEFAULT = `PRI`)
P3	(F)	CLEVEL	Contour level. The visibility of the plotted three-dimensional figure is determined by this value. (DEFAULT = `0.0`)
P4	(F)	IVISBL	Visibility switch: `ABOVE` data values above CLEVEL will be visible; `BELOW` data values below CLEVEL will be visible; (DEFAULT = `ABOVE`)
P5-10		UNUSED	++++++++++++++++++++

END This **END** command terminates the **THREED** command stack.

TRANSFER—Transfers fields from any Cartesian volume contained in an existing disk file to the *current* edit volume in CEDRIC. Transferred fields will be remapped into the Cartesian coordinate system of the existing edit volume. This means that the fields to be transferred must have grid points in common with the *current* edit volume; that is, their grid can only be an integer multiple of the one in the *current* edit volume. Further, the volume from which fields are transferred *must* have the same number of Z-levels as the *current* edit volume. It may be necessary to **RENAME** some edit fields before **TRANSFER** is invoked since this command will overwrite existing fields with the same name as those being transferred. **TRANSFER** is a multi-card command whenever four (4) or more fields are to be transferred from a single Cartesian volume. An **END** card image is **NOT** allowed to terminate this command; instead, **EXTRA** card images are processed until a "blank" field name is encountered.

Command structure:

```
TRANSFERLUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMTRN  NAMTRN  NAMTRN  NAMTRN
        NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN
```

P2	(F)	LUNIT	Fortran logical unit number of the file from which the data fields are to be transferred. LUNIT must be predefined in the Job Control Language.
P3	(A)	NAMVOL	Volume name. If an actual volume name is specified, a forward search will continue until that volume is located. Otherwise, NAMVOL can simply be `NEXT` to tranfer the next volume in the INPUT file within the time range specified by `IBEGTM` and `IENDTM`. (DEFAULT = `NEXT`)
P4	(F)	IBEGTM	Beginning time (HHMMSS) when NAMVOL is set to `NEXT`. (DEFAULT = `000000.0`)
P5	(F)	IENDTM	Ending time (HHMMSS) when NAMVOL is set to `NEXT`. (DEFAULT = `240000.0`)
P6	(A)	IREWND	Rewind option: `YES` rewind the file pointer to the beginning of the file before initiating the search; `NO` do **NOT** rewind the file pointer before initiating the search. (DEFAULT = `NO`)
P7-10	(A)	NAMTRN	Names of the fields to be transferred from the Cartesian volume. If a field to be transferred has the same name as an existing edit volume field, the edit volume field will be overwritten by the transferred field. If exactly four (4) fields are specified, a "blank" card image *must* follow the **TRANSFER** to terminate the processing of field names. Whenever more than four (4) are specified, their names *must* appear on subsequent card images in the following format:

TRANSFER –STACK of EXTRA CARD IMAGES–

P2-10	(A)	NAMTRN	Remaining field names to be transferred. **EXTRA** card images are processed until a "blank" field name is encountered.

UNFOLD—Performs de-aliasing or unfolding of radial velocity measurements within a designated geometrically-windowed region. The available methods are readily tailored to a variety of folding cases. In all methods, a template or reference field U_e is built in some way as an estimate of the "true" or unfolded velocities. The original radial velocities V are then compared to the reference values and put into the proper velocity range using the following scheme.

The true (unfolded) radial velocity U at a (X,Y,Z) grid point is

$$U = V + \kappa V_a; \qquad \kappa = 0, \pm1, \pm2, \ldots \tag{1}$$

where V is the measured quantity which may have been folded and is subject to measurement error, $V_a = 2V_n$ is the ambiguous velocity interval, and κ is the integer factor needed to remove Nyquist folding ambiguities from V. When the measured velocity differs by more than V_n from the value expected at the grid point, the integer folding factor in Eq (1) is non-zero and can be approximated by

$$K = \frac{U_e - V}{V_a} \tag{2a}$$

where U_e is a preliminary estimate of the true radial velocity at the (X, Y, Z) point. The appropriate integer unfolding factor is determined by

$$\kappa = \begin{cases} INT(K + 0.5), & \text{if } K \geq 0 \;, \\ INT(K - 0.5), & \text{if } K \leq 0 \;. \end{cases} \tag{2b}$$

where INT represents truncation toward zero. The quantity U_e comes from the so-called template or reference velocity field, and the field V is unfolded and stored in U, using Eqs. (1) and (2).

Unfolding can be performed by using any one of six available techniques:

AUTO—automatic unfolding. Suspect radial velocity estimates are first decimated, based upon deviation from the local mean. The resultant field is filled at these decimated locations using the Leise filling technique (see the **PATCHER** command). All radial velocities are unfolded using this filled field as a reference.

FORCE(IN/OUT)—forced unfolding. All velocities within the designated spatial window and either **IN**side or **OUT**side of a user-specified velocity range are forced to be within the Nyquist velocity interval of a user-specified reference velocity.

AUTOTEMP—automatic unfolding using a template field. A template field containing reference velocities is specified and used as a "seed" to initiate the procedure at a designated plane. The radial velocities at the designated plane are unfolded using the information in the template field at the corresponding plane as a reference. The resultant unfolded plane is saved and subsequently used as the template for unfolding the next higher and lower planes. This successive-level procedure continues in both directions until all planes within the geometric window have been unfolded. The template field and the field to be unfolded may be the same field.

AUTOFILL—automatic unfolding using a template field with data filling. This procedure is the same as **AUTOTEMP** with one additional feature. After each unfolded plane has been saved, it is data filled using the Leise technique (see the **PATCHER** command) before it is used as a template for the next plane to be unfolded. This ensures that a reference value will be present at every grid point within each successive plane.

AUTOSLOP—automatic unfolding using a template field with data filling and slope preservation in the direction of execution. This procedure is the same as **AUTOFILL** with one additional feature. After two consecutive unfolded planes have been produced, the slope at each location is taken into account when generating the template for the next plane to be unfolded.

FORTEMP—forced template unfolding. A template field is specified, and the velocity information in it is used as a reference for unfolding the input folded velocity field at each corresponding location.

Aliased velocities will be unfolded as many times as are necessary in order to place them within the Nyquist velocity interval of the reference velocity. Only those locations within the designated window will be unfolded. *Note: This is a single card image command.*

Command structure:

```
UNFOLD  NAMOUF  NAMINF  METHOD  P5      P6      P7      P8      IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTO    LOCDIM  MINPTS  MAXDEV  VNYQ    IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  FORCEIN VNYQ    VREF    VMIN    VMAX    IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  FORCEOUTVNYQ    VREF    VMIN    VMAX    IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTOTEMPNAMTPF  VNYQ    ISPCNX  LEVEL   IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTOFILLNAMTPF  VNYQ    ISPCNX  LEVEL   IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTOSLOPNAMTPF  VNYQ    ISPCNX  LEVEL   IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  FORTEMP NAMTPF  VNYQ                    IFIXAX  IWINDO
```

P2 (A) NAMOUF — Name of the output field containing the results from this command. (DEFAULT = The input field will be updated if NAMOUF is "blank".)

P3 (A) NAMINF — Name of the input field to be altered.

P4 (A) METHOD — The method of unfolding. Depending on the `METHOD` specified, the remaining information on the card image will be interpreted accordingly:

`---METHOD (P4) set to AUTO---P5-P10 contain parameterization---`

P5 (F) LOCDIM — Dimension of the local region surrounding each location. LOCDIM must be odd and the region will be assumed (LOCDIM x LOCDIM) in size. (DEFAULT = `3.0`)

P6 (F) MINPTS — Minimum number of points required in the (LOCDIM x LOCDIM) region. At least MINPTS values must be present or the central value is flagged **BAD** or decimated. (DEFAULT = `4.0`)

P7 (F) MAXDEV — Maximum permissible deviation from the local mean for a center value to be retained. (DEFAULT=Nyquist Velocity divided by 2)

P8 (F) VNYQ — Nyquist velocity (M/S) to be used for unfolding. (DEFAULT=Nyquist velocity in the volume header.)

P9 (A) IFIXAX — Axis to be fixed for two-dimensional functions:
- `Z` z-axis fixed, operate in xy-planes;
- `Y` y-axis fixed, operate in xz-planes;
- `X` x-axis fixed, operate in yz-planes.

(DEFAULT = `Z`)

P10 (A) IWINDO — Windowing specification for this command:
- `WINDOW` execute the command only in the current user-set window;
- `FULL` execute the command over the full (X,Y,Z) grid.

(DEFAULT = `FULL`)

`---METHOD (P4) set to FORCEIN---P5-P10 contain parameterization---`

P5	(F)	VNYQ	Nyquist velocity (M/S) to be used for unfolding. (DEFAULT=Nyquist velocity in the volume header.)
P6	(F)	VREF	Reference velocity. All velocities within the designated spatial window and **IN**side the velocity interval (VMIN,VMAX) will be unfolded into the same Nyquist interval as VREF. (DEFAULT = `0.0`)
P7	(F)	VMIN	Minimum value of velocity interval (M/S). (DEFAULT = `0.0`)
P8	(F)	VMAX	Maximum value of velocity interval (M/S). (DEFAULT = `0.0`)
P9	(A)	IFIXAX	Axis to be fixed. (See above description.)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

`---METHOD (P4) set to FORCEOUT--P5-P10 contain parameterization---`

P5	(F)	VNYQ	Nyquist velocity (M/S) to be used for unfolding. (DEFAULT=Nyquist velocity in the volume header.)
P6	(F)	VREF	Reference velocity. All velocities within the designated spatial window and **OUT**side the velocity interval (VMIN,VMAX) will be unfolded into the same Nyquist interval as VREF. (DEFAULT = `0.0`)
P7	(F)	VMIN	Minimum value of velocity interval (M/S). (DEFAULT = `0.0`)
P8	(F)	VMAX	Maximum value of velocity interval (M/S). (DEFAULT = `0.0`)
P9	(A)	IFIXAX	Axis to be fixed. (See above description.)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

`---METHOD (P4) set to AUTOTEMP---P5-P10 contain parameterization---`

P5	(A)	NAMTPF	Name of the template field. This field *must* be specified, and may be the same as the field to be unfolded.
P6	(F)	VNYQ	Nyquist velocity (M/S) to be used for unfolding. (DEFAULT=Nyquist velocity in the volume header.)
P7	(A)	ISPCNX	Context within which the (X,Y,Z) location is defined: `INDEX` coordinates are specified in index space; `DISTANCE` coordinates are specified in distance space; (DEFAULT = `INDEX`)
P8	(F)	LEVEL	Starting plane for the automatic template unfolding. Unfolding will proceed in both a positive and a negative direction from this plane. (DEFAULT=Lowest plane within the spatial window.)
P9	(A)	IFIXAX	Axis to be fixed. (See above description.)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

---METHOD (P4) set to AUTOFILL---P5-P10 contain parameterization---

P5	(A)	NAMTPF	Name of the template field. This field *must* be specified, and may be the same as the field to be unfolded.
P6	(F)	VNYQ	Nyquist velocity (M/S) to be used for unfolding. (DEFAULT=Nyquist velocity in the volume header.)
P7	(A)	ISPCNX	Context within which the starting plane is defined. (See above description.)
P8	(F)	LEVEL	Starting plane for the automatic template unfolding. (See above description.)
P9	(A)	IFIXAX	Axis to be fixed. (See above description.)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

---METHOD (P4) set to AUTOSLOP---P5-P10 contain parameterization---

P5	(A)	NAMTPF	Name of the template field. This field *must* be specified, and may be the same as the field to be unfolded.
P6	(F)	VNYQ	Nyquist velocity (M/S) to be used for unfolding. (DEFAULT=Nyquist velocity in the volume header.)
P7	(A)	ISPCNX	Context within which the starting plane is defined. (See above description.)
P8	(F)	LEVEL	Starting plane for the automatic template unfolding. (See above description.)
P9	(A)	IFIXAX	Axis to be fixed. (See above description.)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

---METHOD (P4) set to FORTEMP---P5-6,P9-10 contain parameterization---

P5	(A)	NAMTPF	Name of the template field. This field *must* be specified. If it is the same as the field to be unfolded, no unfolding will be done.
P6	(F)	VNYQ	Nyquist velocity (M/S) to be used for unfolding. (DEFAULT=Nyquist velocity in the volume header.)
P7-9		UNUSED	++++++++++++++++++++
P9	(A)	IFIXAX	Axis to be fixed. (See above description.)
P10	(A)	IWINDO	Windowing specification for this command. (See above description.)

VALUE—Provides a mechanism by which CEDRIC commands can be parameterized so that the appropriate values may be substituted into the commands at the time of command execution. This command is typically used in conjunction with the **DEFINE** and **EXPAND** commands to facilitate the execution of repetitious multi-command CEDRIC procedures without their being repeated in the script and without having to make parameters explicit. This command permits the user to associate new values with variable names that have been entered as parameters to CEDRIC commands in order to change their numeric values before execution or re-execution of a **DEFINE** block. Only one set of variable assignments may exist at a time containing no more than `100` entries. See P2 for a description of allowed assignment manipulations.

When the **VALUE** list is non-empty and substitution has been activated, parameter fields P2 thru P10 of every subsequent card image are scanned prior to execution. If a particular parameter field is identical character-by-character to a variable name in the **VALUE** list, it will be replaced by the value to which it has been assigned. All eight characters must agree with one another for a replacement to be made. No variable name may be doubly defined and no parameter field will be substituted for more than once.

The user can also assign elements of the ID header to variables names. For example, the Nyquist velocity, which is ID(304) (see Appendix D), could be assigned to a variable name and then that variable could be used in Cedric commands. This way, the user doesn't have to manually enter the Nyquist velocity when it's needed. *Note: This is a single card image command.*

Command structure:

```
VALUE    IOPCOD  NAMVAR           VALUE
```

P2	(A)	IOPCOD	Operation code: `OFF` turns OFF substitutions of values for variables, no additional parameters need to be specified; `ON` turns ON substitution of values for variables, no additional parameters need to be specified; `NONE` deletes **ALL** existing variable assignments, no additional parameters need to be specified; `DELETE` deletes a variable assignment from the list; `SET` adds a variable assignment to the existing list. `SETID` adds a variable assignment to the existing list. In this case, though, the user specifies a header word from which to obtain the value. (DEFAULT = `OFF`)
P3	(A)	NAMVAR	Variable name. This may consist of any string of up to 8 characters or numbers, including decimal points, that are to be replaced in the parameter fields of CEDRIC card images when substitution has been turned `ON`. All eight (8) symbols must agree for a replacement to be made. NAMVAR must be specified whenever `IOPCOD = SET` or `IOPCOD = DELETE`.
P4		UNUSED	++++++++++++++++++++ If this parameter is set to `TO`, it helps remind the user of what is being done when `IOPCOD = SET`.
P5	(A)	VALUE	Value assigned to NAMVAR. It can be any valid ASCII string of up to 8 symbols. This `VALUE` will be substituted into the parameter field of subsequent CEDRIC card images if substitution is turned `ON` and the associated NAMVAR matches that field as it was originally specified on the CEDRIC card image. If the content of this parameter is "ID(x)", where x is an integer value from 1-510, and `IOPCOD = SETID`, that element from the ID header will be the value assigned to the variable. `VALUE` must be specified whenever `IOPCOD = SET` or `IOPCOD = SETID`.
P6-10		UNUSED	++++++++++++++++++++

WINDOW—Defines or deactivates a user-supplied spatial window that may be subsequently invoked for purposes of selective editing and/or display (when the IWINDO variable in several CEDRIC commands is set to `WINDOW`). This command is particularly useful for constraining the execution of commands to small regions without affecting the remainder of the field. However, this option should be exercised cautiously when doing the **FUNCTION** command since the final field may not be derived in the same way everywhere. This command also permits the user to generate graphics on any size grid background so that regions of interest may be examined in detail, and CEDRIC displays can be sized for comparison with displays from other measurement systems. The window specified may be larger or smaller than the *actual* edit volume grid in which the information is contained so long as they overlap in some way. Whenever a new edit file is established using either the **READVOL** or the **SYNTHES** command, the window established by this command will be *reset* to the full (X,Y,Z) grid corresponding to the new edit volume. **WINDOW** differs from the **CREATE** command in that **WINDOW** does not perform any restructuring of the coordinate system, whereas **CREATE** actually establishes a new (X,Y,Z) grid mapping. *Note: This is a single card image command.*

Command structure:

```
WINDOW  ISPCNX  X1      X2      Y1      Y2      Z1      Z2
```

P2	(A)	ISPCNX	Context within which the (X,Y,Z) location is defined: `INDEX` coordinates are specified in index space; `DISTANCE` coordinates are specified in distance space; (DEFAULT = `INDEX`)
P3	(F)	X1	Beginning X-coordinate of the (X,Y,Z) window.
P4	(F)	X2	Ending X-coordinate. (DEFAULT=If either X1 or X2 are unspecified, the X-axis of the window will correspond to the X-axis of the full (X,Y,Z) grid.)
P5	(F)	Y1	Beginning Y-coordinate of the (X,Y,Z) window.
P6	(F)	Y2	Ending Y-coordinate. (DEFAULT=If either Y1 or Y2 are unspecified, the Y-axis of the window will correspond to the Y-axis of the full (X,Y,Z) grid.)
P7	(F)	Z1	Beginning Z-coordinate of the (X,Y,Z) window.
P8	(F)	Z2	Ending Z-coordinate. (DEFAULT=If either Z1 or Z2 are unspecified, the Z-axis of the window will correspond to the Z-axis of the full (X,Y,Z) grid.)
P9-10		UNUSED	++++++++++++++++++++

WRITVOL—Writes the current edit volume out to disk. There are two output formats that can be selected. The first is called pure binary format and is portable, very fast, and compact. This format is recommended for general use. See **Appendix D** for specifics of this format. The second format that files can be created in is netCDF. This format was developed by NASA and the Unidata program at UCAR. It is highly portable. There is also a lot of software around that can directly read this format. Thus, if you want to use Cedric files in other software packages, it might be easier to create them in this format. The pure format can also be used in your own programs, but it would be harder to do. See **Appendix D** for information about that. I/O to netCDF files is slower and the files are bigger, so it is not recommended for general, everyday use. Furthermore, only one volume can be written to netCDF files. With the pure format, up to 10 volumes can be placed in a single file. Writing out to a file that already exists with the netCDF format option selected will cause the contents of the existing file to be overwritten. No appending takes place. Users should exercise caution to avoid accidentally overwriting datasets.

WRITVOL has no effect on the *current* edit volume. It merely copies it to a permanent storage medium for reference at a later time. Also, any number of output volumes can be written using **WRITVOL** and re-accessed if necessary with **READVOL** during the course of a single CEDRIC run. All output positioning options except `BEG` will add volumes to the end of existing files. If `BEG` is specified, the disk file is positioned at the beginning. Any volumes of data that already exist in this file will be lost. Some terminology here is a carry-over from when physical tapes were actually mounted and accessed by CEDRIC. This rarely occurs anymore since tape is usually put onto disk first and then accessed. *Note: This is a single card image command.*

Command structure:

```
WRITVOL LUN      NAMTAP  NAMVOL  IPOS    NFSKIP  FEET    FORMAT
```

P2	(F)	LUN	Fortran logical unit number of the OUTPUT file.
P3	(A)	NAMTAP	An internal CEDRIC name of the output disk file. (DEFAULT = `UNKNOWN`)
P4	(A)	NAMVOL	An internal CEDRIC name of the output volume. This may be any eight (8) character alphanumeric entity. (DEFAULT=The name of the volume as it is referred to in the *current* edit file.)
P5	(A)	IPOS	OUTPUT disk positioning code prior to the write. If the file is PURE binary: `BEG` position at the beginning; `APP` append to the end of the file. (DEFAULT=This parameter must be specified.) If the file is netCDF no positioning options are possible.
P6	(F)	NFSKIP	Number of files to be skipped when `IPOSN = SKI`. (DEFAULT = `0.0`)
P7	(F)	FEET	Number of feet skipped when either `ADD` or `SKI` is specified as the positioning code. This information is available from the output listing of the CEDRIC run which last wrote on the "tape". Although this information is not crucial it does aid CEDRIC in determining when a user attempts to exceed the capacity of a 9-track, 6400 BPI tape. (DEFAULT = `0.0`)
P8	(A)	FORMAT	Output format for the data. There are two possibilities: `PUR` for pure binary files. `CDF` for netCDF files. (DEFAULT = `CDF`)
P9-10		UNUSED	++++++++++++++++++++

WTRANS—Transfers fields from any volume contained on an existing disk file into the *current* edit volume. Transfers are done either into the same field name if it already exists, or into a name carried from the transfer-file header. If no tranlation or windowing is wanted during the transfer, the **TRANSFER** command should be used instead of **WTRANS**. Transferred fields will be remapped into the coordinate system of the existing edit volume under user supplied constraints. These allow for prior translation of the input data locations and selection of the destination region to be modified in cases where the field is already present in the edit volume. Translation *must* be performed so that grid points in the input volume map to coincident locations in the edit volume. Windowing *only* applies to fields that are present in both the input and the edit volumes. It may be necessary to **RENAME** some edit fields before **WTRANS** is invoked in order to avoid or to force the duplication of names depending upon the outcome desired. **WTRANS** is a stack operation and *must* be terminated by an **END** command.

Command structure:

```
WTRANS  LUNIT   NAMVOL  IBEGTM  IENDTM  ISPCNX  XTRANS  YTRANS  ZTRANS  IWINDO
NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN
END
```

P2 (H) LUNIT — Fortran logical unit number of the INPUT file from which the fields are to be transferred. `LUNIT` may be specified in either of two formats:
`R=nnnnnn` where `nnnnnn` is a 6 digit floating point value specifying the Fortran logical unit number. When this format is used the file is repositioned to its beginning prior to the forward search for the requested volume;
`nnnnnnnn` If `R=` is **NOT** present the entire 8 digit field `nnnnnnnn` will be interpreted as a unit number and NO repositioning within the file will be performed.
This parameter *must* be specified and `LUNIT` predefined in the Job Control Language.

P3 (A) NAMVOL — Volume name. If an actual volume name is specified, a forward search will continue until that volume is located. Otherwise, NAMVOL can simply be `NEXT` to tranfer the next volume in the INPUT file within the time range specified by `IBEGTM` and `IENDTM`.
(DEFAULT = `NEXT`)

P4 (F) IBEGTM — Beginning time (HHMMSS) when NAMVOL is set to `NEXT`.
(DEFAULT = `000000.0`)

P5 (F) IENDTM — Ending time (HHMMSS) when NAMVOL is set to `NEXT`.
(DEFAULT = `240000.0`)

P6 (A) ISPCNX — Context within which the (X,Y,Z) translation is to be done:
`INDEX` translations are specified in index space;
`DISTANCE` translations are specified in distance space;
(DEFAULT = `INDEX`)

P7 (F) XTRANS — Input volume translation in X: `NEW X = CURRENT X + XTRANS`.
(DEFAULT = `0.0`)

P8 (F) YTRANS — Input volume translation in Y: `NEW Y = CURRENT Y + YTRANS`.
(DEFAULT = `0.0`)

P9	(F)	ZTRANS	Input volume translation in Z: `NEW Z = CURRENT Z + ZTRANS`. (DEFAULT = `0.0`) NOTE: The `(XTRANS,YTRANS,ZTRANS)` displacements *must* result in coincident locations between the two volumes.
P10	(A)	IWINDO	Windowing specification for **WTRANS** transfer of fields that already exist in the edit volume. This feature applies ONLY to existing fields; newly created fields are ALWAYS transferred in their entirety to the coordinate system of the *current* edit volume: `WINDOW` existing fields will be updated ONLY within the confines of the current user-set window; `FULL` existing fields will be completely replaced. (DEFAULT = `FULL`)

WTRANS	–STACK of EXTRA CARD IMAGES–
P1-10 (A) NAMTRN	Names of the fields to be transferred from the volume to which the disk file has been positioned. The data from each field will be transferred to the *current* edit file according to the conventions previously given.

END This **END** command terminates the **WTRANS** command stack.

Leave this page blank

Leave this page blank

APPENDICES A-F

CEDRIC—Custom Editing and Display of Reduced Information in Cartesian space

BATCH PROCESSOR for UNIX-BASED COMPUTERS

Leave this page blank

Leave this page blank

APPENDIX A

+++++ A Summary listing of all CEDRIC commands +++++
Names of parameters are listed

This is a summary of all CEDRIC commands that have been documented in detail. In this summary, the *Command strructure* is emphasized. Following this summary listing are commands where values or different options have been substituted for the parameter names.

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
BADVAL  BAD
CHANGE  NAMFLD  ISPCNX  XLOC    YLOC    ZLOC    VALUE
CODED   IDEST   IFIXAX  SKIP                                            IWINDO
        NMSPEC  SCLFAC  ZREF    WIDTH   NLFT    NRGT
END
COMMENT IDEST
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
COMEND
CONVERGENAMOUF  NAMU    NAMV    NDER
COORD   ICORD
CREATE  X1      X2      XD      Y1      Y2      YD      Z1      Z2      ZD
CROSS   NAMOUF  NAMREF  NAMLAG  LAGI    LAGJ    IZERO   JZERO   IFIXAX  IWINDO
DEFINE  NAMBLK
COMMENT IDEST
        TEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXTTEXT
COMEND
CREATE  X1      X2      XD      Y1      Y2      YD      Z1      Z2      ZD
BADVAL  BAD
CHANGE  NAMFLD  ISPCNX  XLOC    YLOC    ZLOC    VALUE
CODED   IDEST   IFIXAX  SKIP                                            IWINDO
        NMSPEC  SCLFAC  ZREF    WIDTH   NLFT    NRGT
END
COORD   ICORD
CONVERGENAMOUF  NAMU    NAMV    NDER
CROSS   NAMOUF  NAMREF  NAMLAG  LAGI    LAGJ    IZERO   JZERO   IFIXAX  IWINDO
DEFEND
DIGITAL IDEST   IFIXAX  SKIP                                            IWINDO
        NMSPEC  SCLFAC  NDIG
END
DELETE  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL  NAMDEL
END
EXPAND  NAMBLK
FIELDSETNAMGRP  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD  NINCLD
        NINCLD  NINCLD  NINCLD
FILTER  NAMOUF  NAMINF  METHOD  NSTEP                           IFIXAX  IWINDO
FIXIDS  IWORD   IVALUE  IWORD   IVALUE  IWORD   IVALUE  IWORD   IVALUE
FLTERTH OPTION
FUNCTIONIDMODE  DIREC                                           IFIXAX  IWINDO
        NAMOUF  IFLTYP  NAMFUN  NAMF1   NAMF2   C1      C2      C3      C4
        NAMOUF  C       CONPLANEINFLD   CONSET
END
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ A Summary listing of all CEDRIC commands +++++
Names of parameters are listed

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
GRAPHICSIX*IY*IZIFIXAX  SKIP    ITITLEITITLEITITLEITITLEITITLEITITLE    IWINDO
        CONTOUR NAMFLD  ICONTP  IOPTNS  DIGSCL  ICLDEF  CLMIN   CLMAX   CLINC
        VECTOR  NAMHOR  NAMVER  IVPTYP  IVSPAC  VMS     VKM     IVHEAD  COLOR
        SCATTER NAMHOR  XBEG    XEND    NAMVER  YBEG    YEND    DEGFIT  XCOL
        ALPHA   NMSPEC  SCLFAC  ZREF    WIDTH   NLFT    NRGT    ISPAC
        DIGITAL NMSPEC  SCLFAC  NDIG    ISPAC
        OVERLAY IFPRIM  IFOVLY  IOPTYP  IDASHP  IDASHO  PRICOL  OVLCOL
END
HISTO   IDEST   IFIXAX  SKIP    ITYP    TTITLETTITLETTITLETTITLETTITL   IWINDO
        NMSPEC  SCLFAC  HMIN    HMAX    WIDTH   MEDIAN  COLOR   PERC    DISOPT
END
INTEGR  NAMOUF  NAMINF  INTYPE  METHBC  INIVAL  NAMTOP          ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  U*nnnnnnMETHBC  INIVAL  NAMTOP          ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  D*nnnnnnMETHBC  INIVAL  NAMTOP          ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  V*nnnnnnMETHUP  INIVUP  METHDN  INIVDN  ZLOWER  ZUPPER
LAPLACE NAMOUF  DPDX    DPDY    RELEPS  MAXITR
MASS2   UNEW    VNEW    WNEW    INTYPE  METHBC  INIVAL  CNCRIT  ZLOWER  ZUPPER
        ZINIT   NAMDBZ  ACON    BCON    CCON
PATCHER NAMOUF  NAMINF  METHOD  P5      P6      P7      P8      IFIXAX  IWINDO
PATCHER NAMOUF  NAMINF  FILLCON MAXSTP  MINQAD  MINPTS  DECFLD  IFIXAX  IWINDO
PATCHER NAMOUF  NAMINF  FILLCON3MAXSTP  MINQAD  MINPTS                  IWINDO
PATCHER NAMOUF  NAMINF  FILLALL NDIM                                    IWINDO
PATCHER NAMOUF  NAMINF  DECILOC MAXSTP  MINPTS  MAXDEV          IFIXAX  IWINDO
PATCHER NAMOUF  NAMINF  DECIGLO SIGFAC                                  IWINDO
QUIT
READVOL LUN     NAMVOL  IBEGTM  IENDTM  IREWND
REGRESS IDEST   IFIXAX  SKIP    ITYP    NAMIND  NAMDEP                  IWINDO
RELAXUV UNEW    VNEW    UCUR    VCUR    WAIR    DWFAC   CNCRIT  ZLOWER  ZUPPER
REMAP   ICORD   ANGXAX  XORGIN  YORGIN  ZORGIN  NWGRD   MAXREL
        X1      X2      XD      Y1      Y2      YD      Z1      Z2      ZD
        SPEC    P3      P4      P5      P6      P7      P8      P9      P10
        ROTATE  INX     INY     INZ     OUTX    OUTY    OUTZ
        LINEAR  FIELD
        UNFOLD  FIELD   NYQ
END
RENAME  NAMOLD  TO      NAMNEW  NAMOLD  TO      NAMNEW  NAMOLD  TO      NAMNEW
SAMPLER NAMOUF  NAMU    NAMV    NAMW    XRAD    YRAD    ZRAD    COORD   IWINDO
SHIFTER NAMOUF  NAMINF  NAMTIM  ITANAL  STMDIR  STMSPD  IADTYP  ZLOWER  ZUPPER
STATS   IDEST   IFIXAX  SKIP    NMSPEC  ITYP                            IWINDO
STPLOT  IDEST   IFIXAX  IORENT  MEAN    SIGMA   MIN     MAX     SDFAC   IWINDO
        NMSPEC  STMIN   STMAX
END
SURFACE IDEST   SKIP    ICONAX  XEYMUL  YEYMUL  FEYMUL  LABTYP          IWINDO
        NMSPEC  VALMIN  VALMAX  VALZER
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ A Summary listing of all CEDRIC commands +++++
Names of parameters are listed

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
SYNTHES NSNVOL  NRADS   DTEST1  DTEST2  DTEST3  ITWOEQ  ITANAL  STMDIR  STMSPD
        LUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMVEL  LADTYP  NAMTIM  NADFSW
        INPUT   NAMINP  NAMINP  NAMINP  NAMINP  NAMINP
        OUTPUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT
        LUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMVEL  LADTYP  NAMTIM  NADFSW
        INPUT   NAMINP  NAMINP  NAMINP  NAMINP  NAMINP
        OUTPUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT
        LUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMVEL  LADTYP  NAMTIM  NADFSW
        INPUT   NAMINP  NAMINP  NAMINP  NAMINP  NAMINP
        OUTPUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT  NAMOUT
END
THREED  IDEST   ICONAX  XSTR    YSTR    ZSTR    XEYE    YEYE    ZEYE
        NMSPEC  CLEVEL  IVISBL
END
TRANSFERLUNIT   NAMVOL  IBEGTM  IENDTM  IREWND  NAMTRN  NAMTRN  NAMTRN  NAMTRN
        NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN
UNFOLD  NAMOUF  NAMINF  METHOD  P5      P6      P7      P8      IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTO    LOCDIM  MINPTS  MAXDEV  VNYQ    IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  FORCEIN VNYQ    VREF    VMIN    VMAX    IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  FORCEOUTVNYQ    VREF    VMIN    VMAX    IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTOTEMPNAMTPF  VNYQ    ISPCNX  LEVEL   IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTOFILLNAMTPF  VNYQ    ISPCNX  LEVEL   IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  AUTOSLOPNAMTPF  VNYQ    ISPCNX  LEVEL   IFIXAX  IWINDO
UNFOLD  NAMOUF  NAMINF  FORTEMP NAMTPF  VNYQ                    IFIXAX  IWINDO
VALUE   IOPCOD  NAMVAR          VALUE
WINDOW  ISPCNX  X1      X2      Y1      Y2      Z1      Z2
WRITVOL LUN     NAMTAP  NAMVOL  IPOS    NFSKIP  FEET    FORMAT
WTRANS  LUNIT   NAMVOL  IBEGTM  IENDTM  ISPCNX  XTRANS  YTRANS  ZTRANS  IWINDO
NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN  NAMTRN
END
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Examples of CEDRIC commands +++++

Parameter values rather than Names appear in some instances

This is a summary of all CEDRIC commands where some values or different options have been substituted for the parameter Names to show more specifically how the syntax looks in practice. A complete listing of all the **FUNCTION**s is also included.

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
BADVAL  -1000.0
CHANGE  NAMFLD  INDEX   XX.X    YY.Y    ZZ.Z    0.0
CHANGE  NAMFLD  DISTANCEXX.X    YY.Y    ZZ.Z    BAD
CODED   PRINT   Z       1.                                              FULL
CODED   MICRO   X       1.                                              WINDOW
CODED   PRINT   Y       1.                                              FULL
        PRI     1.0     0.0     5.0     20.     20.
        SEC     1.0     0.0     5.0     20.     20.
        ALL     1.0     0.0     5.0     20.     20.
END
COMMENT PRINT
COMMENT MICRO
COMMENT BOTH
        SUPERCALIFRAGILEISTICEXPEALODOUS  (AND OTHER WORDS OF WISDOM)
COMEND
CONVERGENAMOUF  U       V       3.
CONVERGENAMOUF  U       V       5.
COORD   CART
COORD   COPLANE
CREATE  MIN.X   MAX.X   SPAC.X  MIN.Y   MAX.Y   SPAC.Y  MIN.Z   MAX.Z   SPAC.Z
CREATE  OFF
CROSS   NAMOUF  NAMREF  NAMLAG  LAGI    LAGJ    IZERO   JZERO   Z       FULL
DEFINE  NAMBLK
..... (NORMAL CEDRIC COMMANDS).....
DEFEND  NAMBLK
DELETE  NAMFLD1 NAMFLD2 NAMFLD3 NAMFLD5 NAMFLD6 NAMFLD7 NAMFLD8 NAMFLD9 NAMFLD0
DIGITAL PRINT   Z       1.                                              FULL
DIGITAL MICRO   X       1.                                              FULL
DIGITAL BOTH    Y       1.                                              WINDOW
        PRI     1.0     3.
        SEC     1.0     3.
        ALL     1.0     3.
END
EXPAND  NAMBLK
FIELDSETPRI     NAMFLD1 NAMFLD2 NAMFLD3
FIELDSETSEC     NAMFLD4 NAMFLD5 NAMFLD6 NAMFLD7 NAMFLD8
FILTER  NAMOUF  NAMINF  UNI                                     Z       FULL
FILTER  NAMOUF  NAMINF  HAN                                     Y       FULL
FILTER  NAMOUF  NAMINF  INV                                     X       FULL
FILTER  NAMOUF  NAMINF  TWO                                     Z       WINDOW
FILTER  NAMOUF  NAMINF  LEI     3.0                             Z       FULL
FILTER  NAMOUF  NAMINF  L3D     1.0                             Z       FULL
FILTER  NAMOUF  NAMINF  LEI     NSTEP                           Z       FULL
FILTER  NAMOUF  NAMINF  LEA     NSTEP                           Z       WINDOW
FILTER  NAMOUF  NAMINF  LT3     2.0                             Z       FULL
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Examples of CEDRIC commands +++++
Parameter values rather than Names appear in some instances

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
FIXIDS  IWORD   IVALUE  IWORD   IVALUE  IWORD   IVALUE  IWORD   IVALUE
FUNCTIONNAME    1.0                                             Z       WINDOW
        NAMOUF  P       FUNCNAM NAMFLD1 NAMFLD2 C1      C2      C3      C4
END
FUNCTIONNAME    1.0                                             Z       FULL
        NAMOUF  P       DF/DI   NAMFLD1
        NAMOUF  P       DF/DJ   NAMFLD1
        NAMOUF  P       DDI+DDJ NAMFLD1 NAMFLD2
        NAMOUF  P       DDI-DDJ NAMFLD1 NAMFLD2
        NAMOUF  P       DDJ-DDI NAMFLD1 NAMFLD2
        NAMOUF  P       *D/DI   NAMFLD1 NAMFLD2
        NAMOUF  P       *D/DJ   NAMFLD1 NAMFLD2
        NAMOUF  P       DDI*DDJ NAMFLD1 NAMFLD2
        NAMOUF  P       *       NAMFLD1 NAMFLD2 C1
        NAMOUF  P       SQRT    NAMFLD1
        NAMOUF  P       TENLOG  NAMFLD1         C1      C2
        NAMOUF  P       ORELSE  NAMFLD1 NAMFLD2
        NAMOUF  P       SQ+SQ   NAMFLD1 NAMFLD2
        NAMOUF  P       RHOWGT  NAMFLD1         C1
        NAMOUF  P       +       NAMFLD1 NAMFLD2 C1      C2
        NAMOUF  P       -       NAMFLD1 NAMFLD2 C1      C2
        NAMOUF  P       /       NAMFLD1 NAMFLD2 C1
        NAMOUF  P       RHONORM NAMFLD1         C1
        NAMOUF  P       LINEAR  NAMFLD1         C1      C2
        NAMOUF  P       ABS     NAMFLD1
        NAMOUF  P       ONLYIF  NAMFLD1 NAMFLD2
        NAMOUF  P       ONLYIFC>NAMFLD1 NAMFLD2 C1
        NAMOUF  P       ONLYIFC<NAMFLD1 NAMFLD2 C1
        NAMOUF  P       LN      NAMFLD1         C1      C2
        NAMOUF  P       EXP     NAMFLD1         C1
        NAMOUF  P       POWER   NAMFLD1         C1      C2
        NAMOUF  T       RELPLANENAMFLD1         C1
        NAMOUF  P       XYDIST                  C1      C2
        NAMOUF  P       FUNZ                    C1      C2
        NAMOUF  P       SIN     NAMFLD1         C1      C2
        NAMOUF  P       ONLYIFNONAMFLD1 NAMFLD2
        NAMOUF  P       RANGE                   C1      C2      C3
        NAMOUF  P       COS     NAMFLD1         C1      C2
        NAMOUF  P       RANDOM                  C1      C2
        NAMOUF  P       MAX     NAMFLD1 NAMFLD2
        NAMOUF  P       MIN     NAMFLD1 NAMFLD2
        NAMOUF  P       ATAN2   NAMFLD1 NAMFLD2
        NAMOUF  P       AZ                      C1      C2
        NAMOUF  P       EL                      C1      C2      C3
        NAMOUF  P       CON                     C1
        NAMOUF  P       BAD
        NAMOUF  P       FUNX                    C1      C2
        NAMOUF  P       FUNY                    C1      C2
END
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Examples of CEDRIC commands +++++
Parameter values rather than Names appear in some instances

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
FUNCTIONNAME    1.0                                                     Z       WINDOW
        NAMOUF  P       FUNCNAM NAMFLD1 NAMFLD2 C1      C2      C3      C4
END
FUNCTIONNAME    1.0                                                     Z       FULL
        NAMOUF  P       PROFILE NAMFLD1         C1
        NAMOUF  P       LOGTEN  NAMFLD1         C1      C2
        NAMOUF  P       MODULO  NAMFLD1         C1
        NAMOUF  P       FLOOR   NAMFLD1         C1
        NAMOUF  P       CEILING NAMFLD1         C1
        NAMOUF  P       PROJECT NAMFLD1 NAMFLD2 C1
        NAMOUF  P       BITCOUNTNAMFLD1
        NAMOUF  P       NORMAL                  C1      C2
        NAMOUF  P       PROFMIN NAMFLD1         C1
        NAMOUF  P       PROFMAX NAMFLD1         C1
        NAMOUF  P       IFNOC1C2NAMFLD1 NAMFLD2 C1      C2
        NAMOUF  P       INSIDE  NAMFLD1 NAMFLD2 C1      C2
        NAMOUF  P       OUTSIDE NAMFLD1 NAMFLD2 C1      C2
        NAMOUF  P       COUNT   NAMFLD1         C1
        NAMOUF  P       MEANAX1 NAMFLD1
        NAMOUF  P       MEANAX2 NAMFLD1
        NAMOUF  P       RELI    NAMFLD1         C1
        NAMOUF  P       RELJ    NAMFLD1         C1
        NAMOUF  P       RELIJ   NAMFLD1         C1      C2
END
FUNCTIONNUMBER  -1.0                                                    Z       WINDOW
        NAMOUF  P       1-62    NAMFLD1 NAMFLD2 C1      C2      C3      C4
END
FUNCTIONNAME    1.0                                                     Z       WINDOW
        NAMOUF  C       CONPLANEINFLD   CONSET
END
GRAPHICSIX*IY*IZZ       2.                                                      FULL
        CONTOUR PRI     CONT    0010100 DIGSCL  UNI     0.0     60.0    10.0
        CONTOUR SEC     HAFT    0010100 DIGSCL  NON     0.0     60.0    10.0
        CONTOUR ALL     BOTH    0010100 DIGSCL  NON     0.0     60.0    10.0
        ALPHA   NAMFLD  1.0     0.0     5.0     26      26      -1.0
        DIGITAL NAMFLD  1.0     3.0     -1.0
        VECTOR  NAMHOR  NAMVER  ARROW   -1.     2.0     1.0     FIX
        VECTOR  NAMHOR  NAMVER  STREAM  2.0
        SCATTER NAMHOR  -50.    50.     NAMVER  -50.    50.     0.
        OVERLAY IFPRIM  IFOVLY  C       S       L
END
GRAPHICS 2* 2* 1Z       1.0     RELATIVE: SINGLE RADAR INPUT FIELDS     WINDOW
        VECTOR DFILLMAP FILLMAP ARROW   1.0     5.0     1.0     PRO
        CONTOUR DZDM    FALL    00111000        UNI     -20.0   50.0    5.0
        CONTOUR VRTH    FALL    00011010        UNI     -12.5   12.5    1.0
        CONTOUR VRFL    FALL    00011000        UNI     -12.5   12.5    1.0
END
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Examples of CEDRIC commands +++++
Parameter values rather than Names appear in some instances

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
GRAPHICSMICRO   Z       2.0     RELATIVE: REFLECTIVITY AND CART WINDS   FULL
        VECTOR  U       V       ARROW   -1.0    5.0     1.0     PRO
        CONTOUR DZDM2   FALL    00111010        UNI     -20.0   50.0    5.0
        CONTOUR VRFL2   FALL    00011010        UNI     -12.5   12.5    1.0
        CONTOUR DZDM3   FALL    00111010        UNI     -20.0   50.0    5.0
        CONTOUR VRFL3   FALL    00011010        UNI     -12.5   12.5    1.0
        CONTOUR CONV    FALL    00011000        UNI     -10.5   10.5    1.0
        CONTOUR WUP     FALL    00011000        UNI     -12.5   12.5    1.0
        CONTOUR WDN     FALL    00011000        UNI     -12.5   12.5    1.0
        CONTOUR WVAR    FALL    00011000        UNI     -12.5   12.5    1.0
        CONTOUR WUP-DN  FALL    00011000        UNI     -12.5   12.5    1.0
        CONTOUR HSPD    FALL    00111010        UNI     0.0     20.0    1.0
END
HISTO   MICRO   Z       1.0     VOLUME  RELATIVE: MEASUREMENT STATISTICSFULL
        VRNE    1.0     -20.0   20.0    0.5     ON      BLUE    10.0    DISPLAY
        VRTH    1.0     -20.0   20.0    0.5     ON      BLUE    15.0    DISPLAY
        VRFL    1.0     -20.0   20.0    0.5     ON      BLUE    15.0    DISPLAY
END
HISTO   MICRO   Z       1.0     VOLUME  RELATIVE: SYNTHESIS STATISTICS  FULL
        U       1.0     -10.0   10.0    0.2     ON      BLUE    10.0    DISPLAY
        V       1.0     -10.0   10.0    0.2     ON      BLUE    10.0    DISPLAY
        CONV    1.0     -10.0   10.0    0.2     ON      BLUE    15.0    DISPLAY
        WUP     1.0     -10.0   10.0    0.2     ON      BLUE    15.0    DISPLAY
        WDN     1.0     -10.0   10.0    0.2     ON      BLUE    15.0    DISPLAY
        WVAR    1.0     -10.0   10.0    0.2     ON      BLUE    15.0    DISPLAY
        WUP-DN  1.0     -10.0   10.0    0.2     ON      BLUE    15.0    DISPLAY
END
INTEGR  NAMOUF  NAMINF  NAMTOP  -1.0    YES     CONST   0.0     ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  NAMTOP  1.0     NO      FRACT   1.0     ZLOWER  ZUPPER
INTEGR  NAMOUF  NAMINF  NAMTOP  -1.0    YES     FIELD   FIELD   ZLOWER  ZUPPER
LAPLACE NAMOUF  DFDX    DFDY    .001    400.
COMMENT
        ITERATIVE ITEGRATION COMMAND FOR TWO EQUATION SOLUTION
COMEND
MASS2   UNEW    VNEW    WNEW    D*0.1   FIELD   WIDN    10.01   D=1.0   D=15.0
        D=1.0   ZRAD1   2.5     0.120   0.04
PATCHER NAMOUF  NAMINF  FILLCON 5.      3.      4.              Z       WINDOW
PATCHER NAMOUF  NAMINF  FILLALL 2.                              Z       FULL
PATCHER NAMOUF  NAMINF  DECILOC 3.      4.      NYQ/2           Z       WINDOW
PATCHER NAMOUF  NAMINF  DECIGLO 2.0                             Z       FULL
QUIT
READVOL LUNIT   NEXT    000000. 240000. NO
READVOL LUNIT   NEXT    000000. 240000. YES
READVOL LUNIT   VOLNAME                 NO
REGRESS PRINT   Z       1.      BOTH    VARIND  VARDEP                  WINDOW
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Examples of CEDRIC commands +++++
Parameter values rather than Names appear in some instances

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
REMAP   CART    120.0   XORGIN  YORGIN  ZORGIN  NEWGRID 0.5
        X1.X    X2.X    XD.D    Y1.Y    Y2.Y    YD.D    Z1.Z    Z2.Z    ZD.D
        ROTATE  U       V       W       U       V       W
        LINEAR  DBZCP2
        UNFOLD  VRCP3   NYQ3
        UNFOLD  VRCP4   NYQ4
END
RENAME  NAMOLD  AS      NEMNEW  NAMOLD  AS      NAMNEW  NAMOLD  AS      NAMNEW
SAMPLER NAMOUF  U       V       W       XX.X    YY.Y    ZZ.Z            FULL
SHIFTER NAMOUF  NAMINF  NAMTIME 000000. 0.0     0.0     UNI
STATS   PRINT   Z       1.      PRI     PLANE                           FULL
STATS   MICRO   X       1.      SEC     VOLUME                          FULL
STATS   BOTH    Y       1.      ALL     BOTH                            WINDOW
STPLOT  MICRO   Z       VER     S/M/20. D/S/10. L/-/20. L/+/20. 1.0     WINDOW
        WADJ    0.0     50.0
        PRI     0.0     0.0
        SEC     0.0     0.0
        ALL     0.0     0.0
END
SURFACE MICRO   1       XY      2.      -6.     2.5     FULL    FULL
SURFACE MICRO   1       XY      2.      -6.     2.5     FIELD   WINDOW
SURFACE MICRO   1       XY      2.      -6.     2.5     NONE    FULL
SYNTHES NSNVOL  NRADS   DTEST1  DTEST2  DTEST3  DTEST4  000000. 0.0     0.0
        LUNIT   NAMVOL  000000. 240000. NO      NAMVEL  UNI     NAMTIM  NO
        INPUT   NAMFLD1 NAMFLD2 NAMFDL3 NAMFDL4 NAMFLD5
        OUTPUT  NAMFLD1 NAMFLD2 NAMFDL3 NAMFDL4 NAMFLD5
END
THREED  MICRO   Z       XSTR    YSTR    ZSTR    XEYE    YEYE    ZEYE
        PRI     0.0     ABOVE
        SEC     0.0     BELOW
        ALL     0.0     ABOVE
END
TRANSFERLUNIT   NEXT    000000  240000  NO      NAMFLD1 NAMFLD2 NAMFLD3 NAMFLD4
        NAMFLD5 NAMFLD6 NAMFLD7 NAMFLD8 NAMFLD9
COMMENT
        IF TRANSFERING EXACTLY FOUR FIELDS, MUST INCLUDE A BLANK LINE
COMEND
UNFOLD  NAMOUF  NAMINF  AUTO    3.      4.      NYQ/2   VNYQ    Z       FULL
UNFOLD  NAMOUF  NAMINF  FORCEIN VYNQ    0.0     -100.0  100.0   Z       FULL
UNFOLD  NAMOUF  NAMINF  FORCEOUTVYNQ    0.0     0.0     0.0     Z       FULL
UNFOLD  NAMOUF  NAMINF  AUTOTEMPNAMTPF  VNYQ    INDEX   LEVEL   Z       FULL
UNFOLD  NAMOUF  NAMINF  AUTOTEMPNAMTPF  VNYQ    DISTANCELEVEL   Z       FULL
UNFOLD  NAMOUF  NAMINF  AUTOFILLNAMTPF  VNYQ    INDEX   LEVEL   Z       FULL
UNFOLD  NAMOUF  NAMINF  AUTOFILLNAMTPF  VNYQ    DISTANCELEVEL   Z       FULL
UNFOLD  NAMOUF  NAMINF  AUTOSLOPNAMTPF  VNYQ    INDEX   LEVEL   Z       FULL
UNFOLD  NAMOUF  NAMINF  AUTOSLOPNAMTPF  VNYQ    DISTANCELEVEL   Z       FULL
UNFOLD  NAMOUF  NAMINF  FORTEMP NAMTPF  VNYQ                    Z       WINDOW
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Examples of CEDRIC commands +++++
Parameter values rather than Names appear in some instances

```
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
VALUE   OFF
VALUE   ON
VALUE   NONE
VALUE   DELETE  VALUE
VALUE   SET     NAMVAR  TO      VALUE
WINDOW  FULL
WINDOW  INDEX   MIN.X   MAX.X   MIN.Y   MAX.Y   MIN.Z   MAX.Z
WINDOW  DISTANCEMIN.X   MAX.X   MIN.Y   MAX.Y   MIN.Z   MAX.Z
WRITVOL LUNIT   UNKNOWN NAMVOL  BEG     0.      0.0
WRITVOL LUNIT   NAMTAP  NAMVOL  ADD     0.      0.0
WRITVOL LUNIT   NAMTAP  NAMVOL  SKI     0.      0.0
WRITVOL LUNIT   NAMTAP  NAMVOL  APP     0.      0.0
WTRANS  LUNIT   NAMVOL  HHMMSS  HHMMSS  INDEX   XTRANS  YTRANS  ZTRANS  FULL
NAME1   NAME2   NAME3   NAME4   NAME5   NAME6   NAME7   NAME8   NAME9   NAME10
END
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

Leave this page blank

Leave this page blank

APPENDIX B

TABLE OF CEDRIC FUNCTIONS

This appendix contains a table of all the available CEDRIC functions. The `FUNCTION` command is used to group together and execute any sequence of these functions. This enables the user to algebraically manipulate and to combine data fields. When a data field is operated on by a function each individual value within the designated window is altered. Functions may be referenced either by name or by number.

The following abbreviations are employed in the function definitions:

F_1 —Field number 1
F_2 —Field number 2
C_1 —Constant number 1
C_2 —Constant number 2
C_3 —Constant number 3
∂ —the partial derivative
$+$ —the add operator
$-$ —the subtract operator
$/$ —the divide operator
e —the exponential function, $e = 2.7182818...$
$\sqrt{(\ )}$ —square root of a quantity
$(\ )^2$ —quantity squared
$Mean$ —the mean of all values of a field within an (I,J) plane
Min —the minimum of all values of a field within an (I,J) plane
Max —the minimum of all values of a field within an (I,J) plane
$Good$ —count the number of good ($\neq$ BAD) values of a field within an (I,J) plane
MOD —modulo of a quantity with respect to some reference,
—where $MOD(F, C) = 0$ when $F = C$
BAD —missing data flag
X —The X–distance (km) associated with a Cartesian grid point
Y —The Y–distance (km) associated with a Cartesian grid point
Z —The Z–distance (km) associated with a Cartesian grid point

An example of `FUNCTION`s to compute the advective change of f, where `FINP` $= f$ and `FADV` $= \mathbf{V} \cdot \nabla f$.

$$\mathbf{V} \cdot \nabla f = u\partial f/\partial x + v\partial f/\partial y + w\partial f/\partial z\,.$$

```
FUNCTIONNAME    1.0                                                  Z       FULL
        UPPER   T       RELPLANEFINP    1.0
        LOWER   T       RELPLANEFINP    -1.0
        DF/DZ   P       -       UPPER   LOWER   0.5     0.5
END
FUNCTIONNAME    1.0                                                  Z       FULL
        DF/DX   T       DF/DI   FINP
        U*DF/DX T       *       U       DF/DX
        DF/DY   T       DF/DJ   FINP
        V*DF/DY T       *       V       DF/DY
        W*DF/DZ T       *       W       DF/DZ
        H*DF    T       +       U*DF/DX V*DF/DY 1.0     1.0
        FADV    P       +       W*DF/DZ H*DF    1.0     1.0
END
```

CEDRIC FUNCTIONS

FUNCTION #	NAME	Description of the Function	ELEMENTS #F	#C
1	DF/DI	$\partial F_1/\partial I$, where I refers to the current I-axis	1	0
2	DF/DJ	$\partial F_1/\partial J$, where J refers to the current J-axis	1	0
3	DDI+DDJ	$(\partial F_1/\partial I) + (\partial F_2/\partial J)$, where (I,J) refer to the current non-fixed axes	2	0
4	DDI-DDJ	$(\partial F_1/\partial I) - (\partial F_2/\partial J)$, where (I,J) refer to the current non-fixed axes	2	0
5	DDJ-DDI	$(\partial F_1/\partial J) - (\partial F_2/\partial I)$, where (I,J) refer to the current non-fixed axes	2	0
6	*D/DI	$F_1(\partial F_2/\partial I)$, where I refers to the current I-axis	2	0
7	*D/DJ	$F_1(\partial F_2/\partial J)$, where J refers to the current J-axis	2	0
8	DDI*DDJ	$(\partial F_1/\partial I)(\partial F_2/\partial J)$, where (I,J) refer to current non-fixed axes	2	0
9	*	$C_1 F_1 F_2$	2	1
10	SQRT	$\sqrt{F_1}$	1	0
11	TENLOG	Convert from dB scale to linear scale: $C_1 10^{C_2 F_1}$	1	2
12	ORELSE	F_1, if $F_1 \neq$ BAD; otherwise F_2, if $F_2 \neq$ BAD	2	0
13	SQ+SQ	$F_1^2 + F_2^2$	2	0
14	RHOWGT	Density weighting function: $F_1 C_1 e^{-C_2 Z}$	1	2
15	+	$C_1 F_1 + C_2 F_2$	2	2
16	-	$C_1 F_1 - C_2 F_2$	2	2
17	/	$C_1 F_1 / F_2$	2	1
18	RHONORM	Inverse of density weighting function: $F_1/(C_1 e^{-C_2 Z})$	1	2
19	LINEAR	$C_1 F_1 + C_2$	1	2
20	ABS	$\mid F_1 \mid$	1	0
21	ONLYIF	F_1; only if $F_2 \neq$ BAD	2	0
22	ONLYIFC>	F_1; only if $C_1 > F_2$; i.e. $F_2 < C_1$	2	1
23	ONLYIFC<	F_1; only if $C_1 < F_2$; i.e. $F_2 > C_1$	2	1

CEDRIC FUNCTIONS

FUNCTION # NAME	Description of the Function	ELEMENTS #F	#C
24 LN	Logarithm to the base e: $C_1 ln(C_2 F_1)$	1	2
25 EXP	$C_1 e^{F_1}$	1	1
26 POWER	$C_1 F_1^{C_2}$	1	2
27 RELPLANE	F_1 at (current level + C_1)	1	1
28 XYDIST	Horizontal distance of (X,Y) locations from (C_1, C_2): $H(X,Y) = \left[(X - C_1)^2 + (Y - C_2)^2\right]^{1/2}$	0	2
29 FUNZ	Linear function of height Z: $C_1 Z + C_2$	0	2
30 SIN	Sine function: $sin(C_1 F_1 - C_2)$	1	2
31 ONLYIFNO	F_1; only if $F_2 = \mathtt{BAD}$	2	0
32 RANGE	Slant range of (X,Y,Z) locations from (C_1, C_2, C_3) $R(X,Y,Z) = \left[(X - C_1)^2 + (Y - C_2)^2 + (Z - C_3)^2\right]^{1/2}$	0	3
33 COS	Cosine function: $cos(C_1 F_1 - C_2)$	1	2
34 RANDOM	Uniformly distributed random variable in the interval $\left[C_1, C_2\right]$	0	2
35 MAX	Maximum value of two functions at each (I,J) grid point: $Max\left[F_1, F_2\right]$	2	0
36 MIN	Minimum value of two functions at each (I,J) grid point: $Min\left[F_1, F_2\right]$	2	0
37 ATAN2	Arc tangent function: $Tan^{-1}(F_2/F_1)$	2	0
38 AZ	Azimuth angle of (X,Y) grid points relative to (C_1, C_2): $Az(X,Y) = Tan^{-1}\left[(X - C_1)/(Y - C_2)\right]$	0	2
39 EL	Elevation angle of (X,Y,Z) grid points relative to (C_1, C_2, C_3): $El(X,Y,Z) = Tan^{-1}\left[(Z - C_3)/\left[(X - C_1)^2 + (Y - C_2)^2\right]^{1/2}\right]$	0	3
40 CON	Set the output field to the constant C_1	0	1
41 BAD	Set the output field to BAD, the missing data flag	0	0
42 FUNX	Linear function of distance X: $C_1 X + C_2$	0	2
43 FUNY	Linear function of distance Y: $C_1 Y + C_2$	0	2
44 PROFILE	Profile of the mean value: $C_1\ Mean\left[F_1\right]$	1	1

CEDRIC FUNCTIONS

FUNCTION # NAME	Description of the Function	ELEMENTS #F	#C
45 LOGTEN	Convert from linear to dB (log base 10) scale: $C_1 log(C_2 F_1)$	1	2
46 MODULO	Modulo function: $MOD(F_1, C_1)$	1	1
47 FLOOR	F_1, only if $F_1 > C_1$; otherwise C_1	1	1
48 CEILING	F_1, only if $F_1 < C_1$; otherwise C_1	1	1
49 PROJECT	Projection of (U,V) components onto a plane whose West to East orientation is $Az = C_1$, and $(U,V) = (F_1, F_2)$. If $C_1 = X$ or is "blank", the X-axis angle will be used. If $C_1 = Y$, the Y-axis angle will be used.	2	1
50 BITCOUNT	Counts the number of "1" (non-zero) bits contained in each datum of F_1	1	0
51 NORMAL	Normally distributed random variable: mean value of C_1 and a standard deviation of C_2	0	2
52 PROFMIN	Profile along the current fixed axis of the minimum value of F_1: $C_1\ Min[F_1]$	1	1
53 PROFMAX	Profile along the current fixed axis of the maximum value of F_1: $C_1\ Max[F_1]$	1	1
54 IFNOC1C2	F_1, if $F_2 \neq C_1$; otherwise C_2	2	2
55 INSIDE	F_1, if $C_1 \leq F_2 \leq C_2$	2	2
56 OUTSIDE	F_1, if $F_2 < C_1$ or $F_2 > C_2$	2	2
57 COUNT	Constant times number of good values in F_1: $C_1\ Good[F_1]$	1	1
58 MEANAX1	For each J, store the mean value of F_1 along the i-axis at each I-location of F_{out}: $F_{out}(I,J) = \sum_i^M F_1(i,J)/M$	1	0
59 MEANAX2	For each I, store the mean value of F_1 along the j-axis at each J-location of F_{out}: $F_{out}(I,J) = \sum_j^N F_1(I,j)/N$	1	0
60 RELI	$F_1(I + C_1, J)$	1	1
61 RELJ	$F_1(I, J + C_1)$	1	1
62 RELIJ	$F_1(I + C_1, J + C_2)$	1	2

63 LAT	Compute latitude of each grid point using C_1 and C_2 as latitude and longitude of origin. If C_1 and C_2 are missing, the latitude and longitude of the origin will be obtained from the housekeeping.	0	2
64 LON	Compute longitude of each grid point using C_1 and C_2 as latitude and longitude of origin. If C_1 and C_2 are missing, the latitude and longitude of the origin will be obtained from the housekeeping.	0	2
65 SDEVAX1	Compute standard deviation of F_1 along I axis, holding J and K fixed, for all J and K. Result is a field of standard deviations. For a given (J,K), all I points will have the same value (the standard deviation along that axis).	1	0
66 SDEVAX2	Compute standard deviation of F_1 along J axis, holding I and K fixed, for all I and K. Result is a field of standard deviations. For a given (I,K), all J points will have the same value (the standard deviation along that axis).	1	0

APPENDIX C

EFFECTIVE USE OF COLOR GRAPHICS IN CEDRIC

The color graphics options in CEDRIC ultimately should make it easier for scientists to understand the physical meaning and content of plots and data sets. One of the most useful options is with the CONTOUR subcommand of the GRAPHICS card. Here users can turn on color area filling of contour levels. The result is a plot that is much easier to understand. Users turn on the area filling with the P4 parameter. To emphasize the different levels further, we recommend that black contour lines be drawn in with the color area filling. Options to overlay colored contour lines or colored vectors on color area filled plots are also useful. We suggest using white vectors or white contour lines for the overlayed fields. Other colors can change when overlayed on various other colors. For instance, yellow contour lines look yellow on top of grey, but look black on top of blue. See the main part of this manual for the details about invoking the color options.

When a CEDRIC run with graphics commands is completed, the plots are contained in a metacode file called "gmeta" in the directory that CEDRIC was executed from. Many things can be done with this file to view its contents. One tool for viewing the metacode file is a workstation running some kind of windowing system like X windows. The file should be transported from the Cray to the workstation via the network utilities "rcp" or "ftp". Once on the workstation, the commands of interest are "ctrans" and "idt". These programs are part of the NCAR Graphics package. Both of these commands pop up a window that allows the user to view the various frames in the metacode file. The difference between "idt" and "ctrans" is that "idt" allows the user to jump around to random frames in the metacode file while with "ctrans", the user can only step forward, one frame at a time in the file.

One addition to the Graphics options is the option to fill between contour levels with shades of grey. These plots are especially useful for getting copies on a postscript printer. If you have a metacode file with grey area filled contours, use "ctrans" with the "-d ps.color" option to create a postscript file that can be printed on a black and white postscript laser printer.

Producing hard copies of the metacode file on "shavano" (Cray-YMP) is also an option. One can produce color microfilm and color slides directly from the graphics file. The basic command used is "sendtg" and can be executed from the job script that is run on the Cray. To produce color microfilm, we suggest:

```
sendtg gmeta macr=viewercl titl='XXXXXXX' qual=better
```

To produce color slides, use:

```
sendtg gmeta macr=slidesclsq mnt=yes qual=better
```

For detailed information about processing metacode files on the Cray, contact the consulting office in the Scientific Computing Division at NCAR.

Leave this page blank

Leave this page blank

APPENDIX D

CEDRIC CARTESIAN FILE DESCRIPTION

CEDRIC has used several different file structures throughout its history. The format currently in use has no physical record structure. It is written using the standard C input-output libraries and is a byte stream. Information at the beginning of the file reveals how many logical CEDRIC volumes are contained in the file and what their starting byte addresses are relative to the beginning of the file. CEDRIC files are meant to be portable to most machines in use in the scientific community. The basic requirements are that the machine have C on it and that it represent integers using two's complement. Several words at the beginning of the file indicate the byte ordering convention used by the machine that wrote the file. The machine reading the file will do any byte swapping necessary.

Below is a very brief description of the structure of CEDRIC files. It is not intended that this information be detailed enough that any routines can be written to read and write CEDRIC volumes. We can provide either guidance in that regard or an input-output software package.

FILE—a file consists of 1 or more VOLUMEs. At the beginning of the file is information about the number of VOLUMEs in the file, their byte addresses, and other miscellaneous information.

VOLUME—each volume contains a 3-dimensional set of radar FIELDs mapped to an ordered (X,Y,Z) or (X,Y,C) coordinate system. Volumes are associated with a fixed time of day and may contain a field with temporal information corresponding to each individual location. VOLUMEs are preceded by a 510-word logical record containing header information describing the characteristics of the volume. Within a VOLUME, the data corresponding to each radar FIELD is organized in LEVELs associated with a constant height (Z) or constant coplane angle (C).

LEVEL—A constant Z- or C-level contains the data from each FIELD at that height. LEVELs are preceded by a 10-word logical record containing header information describing the characteristics of the LEVEL. FIELD order is indicated in the VOLUME header. This header is described on the following pages. Within a LEVEL, the data from each field is contiguous; that is, all the data from Field 1 followed by all the data from Field 2 and so forth.

FIELD—Each field contains the information for a single parameter such as radial velocity, reflectivity, U-component, V-component, and so forth. Within a LEVEL, the data from each radar FIELD is organized as a two- dimensional FORTRAN array such that the lower left corner corresponds to the (1,1) element and the upper right corner corresponds to the (M,N) element; where M increases along the X-axis and N increases along the Y-axis. Each element of a FIELD is a 16-bit integer word.

ILLUSTRATION OF FILE STRUCTURE

```
FILE ...
   ·   VOLUME 1...
   ·          ·          LEVEL 1...
   ·          ·              ·        FIELD 1...
   ·          ·              ·            .
   ·          ·              ·            .
   ·          ·              ·            .
   ·          ·              ·            .
   ·          ·              ·            .
   ·          ·              ·            .
   ·          ·              ·        FIELD last
   ·          ·          LEVEL last
   ·          ·
   ·   VOLUME last
END of FILE
```

CEDRIC VOLUME and LEVEL HEADERS

The header for CEDRIC volumes consists of 510, 16-bit words that are to be interpreted as either two ASCII alphanumeric characters (A) or as signed integers (2's complement). Several of the integer values are scaled so that their true representation will be their integer value times or divided by this scaling factor. The two general scaling factors are: CF for angle scaling (a value of 64) and SF for general scaling (a value of 100). Header words that are updated by CEDRIC are listed as "CD"; those that are generated during interpolation using SPRINT are listed as "SP".

WORDS	TYPE	SCALING	UPDATES	DESCRIPTION OF CONTENTS
001-004	A		SP,CD	Tape catalogue number (6 char) or disk file name (8 char, 2 per word)
005-006	A		SP,CD	Program name (4 char, 2 per word)
007	A		SP,CD	Program version (2 char, 2 per word)
008-009	A		SP,CD	Project name (4 char, 2 per word)
010-012	A		SP,CD	Scientist name (6 char, 2 per word)
013-015	A		SP	Radar station or data origin (6 char, 2 per word)
016-017	A		SP	Output coordinate system (CRT, ELEV, or CPL) (4 char, 2 per word)
018-020	A		SP	Tape catalogue number (6 char, 2 per word); depends on computer installation
021-023	I	None	SP	Beginning date of radar volume (YYMMDD)
024-026	I	None	SP	Beginning time of radar volume (HHMMSS)
027-029	I	None	SP	Ending date of radar volume (YYMMDD)
030-032	I	None	SP	Ending time of radar volume (HHMMSS)
033-034	I	None	SP,CD	Coordinate origin: Latitude (DEG-MIN)
035	I	(*SF)	SP,CD	Coordinate origin: Latitude (SEC)
036-037	I	None	SP,CD	Coordinate origin: Longitude (DEG-MIN)
038	I	(*SF)	SP,CD	Coordinate origin: Longitude (SEC)
039	I	None	SP,CD	Z-coordinate of the origin: Height in meters (MSL); always 0.0
040	I	(*CF)	SP,CD	Degrees clockwise from North to positive X-axis
041-042	I	(*SF)	SP,CD	X,Y-coordinates of the origin (km)
043-044	A		SP,CD	Time zone (4 char,2 per word)
045-047	A		SP,CD	Job identification number (6 char, 2 per word)
048-050	A		SP,CD	Submitter's name (6 char, 2 per word)
051-054	A		SP,CD	Date of program run (MMDDYY); (8 char, 2 per word)
055-058	A		SP,CD	Time of program run (HHMMSS); (8 char, 2 per word)
059		Unused		
060	I	None		Tape edition number (Incremented by Edit2D)
061	I	None	SP,CD	Length of header record (always 510)
062	A		SP,CD	Computer installation (2 char, 2 per word)
063	I	None	SP,CD	Number of bits per data value (always 16)
064	I	None	SP,CD	Blocking mode (always 2)
065	I	None	SP,CD	Block size or physical record length (always 3200)
066	A		SP	Data set: subsectioned = "SU" or original = "OR"
067	I	None	SP,CD	Missing data flag (always -32768)
068	I	None	SP,CD	General scaling factor: SF = 100
069	I	None	SP,CD	Angle scaling factor: CF = 64
070		Unused		

WORDS	TYPE	SCALING	UPDATES	DESCRIPTION OF CONTENTS
071-074	A		SP	Input file label 1 (8 char, 2 per word)
075-078	A		SP	Input file label 2 (8 char, 2 per word)
079-082	A		SP	Input file label 3 (8 char, 2 per word)
083-086	A		SP	Input file label 4 (8 char, 2 per word)
087-090	A		SP	Input file label 5 (8 char, 2 per word)
091-094	A		SP	Input file label 6 (8 char, 2 per word)
095		Unused		
096	I	None	SP,CD	Number of data records per field per plane
097	I	None	SP,CD	Number of data records per plane
098	I	None	SP,CD	Number of data records per volume, excluding all headers
099	I	None	SP,CD	Total number of records per volume, including all headers
100	I	None	SP,CD	Total number of records per volume, excluding level headers
101-104	A		SP,CD	Volume scan designation (8 char, 2 per word)
105		Unused		
106	I	None	SP,CD	Number of planes in volume scan
107	I	(/SF)	SP	Volume scanned (km^3)
108	I	(/SF)	SP	Total number of sampling points
109	I	(*SF)	SP	Average sampling density (sample point per km^3)
110	I	None	SP	Number of samples integrated in time series
111	I	None	SP,CD	Physical volume number within disk file
112-115		Unused		
116-118	I	None	CD	Beginning date of the volume (YYMMDD)
119-121	I	None	CD	Beginning time of the volume (HHMMSS)
122-124	I	None	CD	Ending date of the volume (YYMMDD)
125-127	I	None	CD	Ending time of the volume (HHMMSS)
128	I	None		Volume Time (Sec)
129	I	None		Index Number-time (4)
130-131		Unused		
132-133	I	(*SF)		Minimum and maximum ranges of the volume scanned (km)
134	I	None	SP	Average number of gates per beam
135	I	None	SP	Average spacing of gates (m)
136-137	I	None	SP	Minimum and maximum number of gates
138		Unused		
139	I	None	SP	Index number for range (always 1)
140-141		Unused		
142-143	I	(*CF)		Clockwise minimum and maximum azimuths of the sweep (deg)
144	I	None		Average number of beams per scanned sweep
145	I	(*CF)		Average increment between beams per sweep (deg)
146-147	I	None		Minimum and maximum number of beams per sweep
148	I	None		Number of steps per beam (average over scanned volume)
149	I	None		Index number for azimuth angle (always 2)
150		Unused		

WORDS	TYPE	SCALING	UPDATES	DESCRIPTION OF CONTENTS
151	A		SP	Coplane (CO) or ppi (sector scan) (PP) flag
152-153	I	(*CF)	SP	Minimum and maximum elevation angle of the volume scanned (deg)
154	I	None	SP	Number of elevation angles scanned
155	I	(*CF)	SP	Average elevation angle between sweeps (deg)
156	I	(*CF)	SP	Average elevation angle (deg)
157	I	None	SP	Scan-direction indicator (+1 = from lowest to highest elevation angle)
158	I	(*CF)	SP	Baseline angle (CW from true North, coplane only)
159	I	None	SP	Index number for the angle scanned, either coplane or elevation (always 3)
160-161	I	(*SF)	SP,CD	X-coordinate; minimum and maximum values (km)
162	I	None	SP,CD	X-coordinate: number of grid points
163	I	None	SP,CD	X-coordinate: grid spacing (m)
164	I	None	SP,CD	X-coordinate: index number (0,1,2,3,4)
165-166	I	(*SF)	SP,CD	Y-coordinate; minimum and maximum values (km)
167	I	None	SP,Cd	Y-coordinate: number of grid points
168	I	None	SP,CD	Y-coordinate: grid spacing (m)
169	I	None	SP,CD	Y-coordinate: index number (0,1,2,3,4)
170-171	I	(*SF)	SP,CD	Z-coordinate; minimum and maximum values (m)
172	I	None	SP,CD	Z-coordinate: number of grid points
173	I	None	SP,CD	Z-coordinate: grid spacing (m)
174	I	None	SP,CD	Z-coordinate: index number (0,1,2,3,4)
175	I	None	SP,CD	Number of fields (Maximum of 25)
176-179	A		SP,CD	Name of first field (8 char, 2 per word)
180	I	None	SP,CD	Scaling factor for first field, actual value = value in file divided by scaling factor
181-184	A		SP,CD	Name of second field (8 char, 2 per word)
185	I	None	SP,CD	Scaling factor for second field, actual value = value in file divided by scaling factor
+				
+				
296-299	A		SP,CD	Name of 25th field (8 char, 2 per word)
300	I	None	SP,CD	Scaling factor for 25th field, actual value = value in file divided by scaling factor
301	I	None	SP,CD	Number of grid points per field per plane
302	I	None	SP,CD	Number of landmarks
303	I	None	SP,CD	Number of radars
304	I	(*SF)	SP,CD	Nyquist velocity (if word 303 = 1)
305	I	(*SF)	SP,CD	Radar constant (if word 303 = 1)
306-308	A		SP,CD	Name of landmark #1, must be “ORIGIN” (6 char, 2 per word)
309-310	I	(*SF)	SP,CD	Landmark #1: (X,Y) coordinates (km), always (0,0)
311	I	None	SP,CD	Landmark #1: Z-coordinate (m), always (0)

WORDS	TYPE	SCALING	UPDATES	DESCRIPTION OF CONTENTS
312-314	A		SP,CD	Name of landmark #2; name of radar if word 303 = 1 (6 char, 2 per word)
315-316	I	(*SF)	SP,CD	Landmark #2 (X,Y) coordinates (km)
317	I	None	SP,CD	Landmark #2: Z-coordinate (m)
318-320	A		SP,CD	Name of landmark #3; name of 2nd radar if word 303 > 1 (6 char, 2 per word)
321-322	I	(*SF)	SP,CD	Landmark #3 (X,Y) coordinates (km)
323	I	None	SP,CD	Landmark #3: Z-coordinate (m)
+	A			
+	I			
+	I			
342-344	A		SP,CD	Name of landmark #7; name of landmark if word 303 > 5 (6 char, 2 per word)
345-346	I	(*SF)	SP,CD	Landmark #7 (X,Y) coordinates (km)
347	I	None	SP,CD	Landmark #7: Z-coordinate (m)
348-395				Landmarks 8-15
396-510				—RESERVED FOR PROGRAM USE—

The header for CEDRIC levels consists of 10, 16-bit words that are to be interpreted as either two ASCII alphanumeric characters (A) or as signed integers (2's complement). Several of the integer values are scaled so that their true representation will be their integer value times or divided by this scaling factor. The general scaling factor is SF for general scaling (a value of 100). Header words that are updated by CEDRIC are listed as "CD"; those that are generated during interpolation using SPRINT are listed as "SP".

WORDS	TYPE	SCALING	UPDATES	DESCRIPTION OF CONTENTS
001	A		SP,CD	Always = "LE"
002	A		SP,CD	Always = "VE"
003	A		SP,CD	Always = "L "
004	I	(*SF)	SP,CD	Coordinate of the plane (or level); (X, Y, or Z km)
005	I	None	SP,CD	Plane (or level) number within the volume
006	I	None	SP,CD	Number of fields
007	I	None	SP,CD	Number of grid points per plane (or level)
008	I	None	SP,CD	Number of records per field
009	I	None	SP,CD	Number of records per plane
010		Unused		

Leave this page blank

Leave this page blank

APPENDIX E
RUNNING CEDRIC ON THE CRAY-YMP AT NCAR

In order to use CEDRIC on the NCAR CRAY computer, the user must construct an input deck with the appropriate Unix shell script commands for accessing and disposing any datasets that are to be created or manipulated. The user is assumed to be reasonably familiar with the Unix operating system and especially its commands and their structures. Those who are not familiar with Unix should contact the Scientific Computing Division (SCD) at NCAR for documentation or assistance. Alternatively, an excellent reference on writing Unix Bourne shell scripts is:

Kernighan, B.W. and R. Pike, 1984: *The Unix Programming Environment.* Prentice Hall.

The following is a complete template for setting up a CEDRIC run,including the actual CEDRIC command structure. Please refer to it for the remainder of this discussion. The first part of the script contains resource declarations. The amount of CPU time, the memory needed, and so forth are specified with the "QSUB" commands. Normally, a line that begins with a "#" is treated as a comment by the Cray. However, one exception is when a line starting with a "# " is followed by "QSUB". In that case, the parameters immediately following the "QSUB" are passed to the batch processing utility as resource requests.

Following the resource declarations are the input dataset acquisitions. Each dataset or file is acquired from the Mass Store System (MSS) with the "msread" command and placed on the Cray disks. These disk files must then be linked to FORTRAN unit numbers that will be used later when CEDRIC executes. The CEDRIC executable code must also be acquired from the MSS. Output volumes need to be linked as well before the execution of CEDRIC.

After the listing of the CEDRIC input card images comes the actual execution of the code. After execution, any output files are sent to the MSS with the "mswrite" command. There are many options for dealing with the graphics metacode file created by CEDRIC as well. The graphics file produced by CEDRIC always has the name "gmeta". This file can be sent to the Text and Graphics System (TAGS) for laser printing or for development on microfilm or microfiche. The file can also be transferred via the Internet to remote workstations for viewing. Remote viewing is accomplished using "ctrans" or "idt" that are part of the NCAR Graphics package.

```
++++++++++A COMPLETE SCRIPT FOR RUNNING CEDRIC ON THE NCAR CRAY-YMP++++++++++

# QSUB -q econ          # put in economy queue
# QSUB -s /bin/sh       # use the Bourne shell
# QSUB -lT 5:00         # upper limit on CPU time for this script (here 5 min.)
# QSUB -lM 7Mw          # internal Cray memory needed to run Cedric
# QSUB -lF 40Mw         # upper limit on the sum of the sizes of files
#                         created by this script
#start accounting
ja

# move to large temporary directory since permanent directories are too small
cd $TMPDIR

# acquire the cedric executable; no compiling necessary
msread -f BI cedric.e /ANDERSNB/CEDRIC/cedric.e || exit
chmod +x cedric.e

# acquire input dataset from Mass Store and linked to fortran unit
# CHANGE the filenames here to yours
msread wdaft21 /TUTTLE/TREC/CART/CP3/25JUN01 || exit
ln -s wdaft21 fort.21

# link output volume to a fortran unit
ln -s wdaft31 fort.31
```

```
# the following is the cedric input being redirected to cedric.inp
# input is terminated by 'EOFA'
cat > cedric.inp << EOFA
COMMENT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
COMEND
DEFINE
FUNCTIONNAME    1.0                                             Z       WINDOW
        UEDIT   P       ONLYIFC<UTREC   VVAR    0.5
        VEDIT   P       ONLYIFC<VTREC   VVAR    0.5
        UEDIT   P       ONLYIFC>UEDIT   VELVAR  8.0
        VEDIT   P       ONLYIFC>VEDIT   VELVAR  8.0
END
PATCHER UEDIT   UEDIT   DECILOC 3.0     4.0     1.5
PATCHER UEDIT   UEDIT   DECILOC 3.0     4.0     1.5
PATCHER VEDIT   VEDIT   DECILOC 3.0     4.0     1.5
PATCHER VEDIT   VEDIT   DECILOC 3.0     4.0     1.5
PATCHER UEDIT   UEDIT   FILLCON 5.0     3.0     4.0
PATCHER VEDIT   VEDIT   FILLCON 5.0     3.0     4.0
FILTER  UEDIT   UEDIT   TWO
FILTER  VEDIT   VEDIT   TWO
WINDOW  DISTANCE                                2.0     3.0
GRAPHICS        Z       1.0             MAGTHR=0.5                      WINDOW
        VECTOR  UEDIT   VEDIT   STREAM  1.0     10.0    5.0     PRO
END
WRITVOL 31.0                    APP
DEFEND
READVOL 21.     NEXT    100000.
EXPAND
READVOL 21.     NEXT    100000.
EXPAND
READVOL 21.     NEXT    100000.
EXPAND
READVOL 21.     NEXT    100000.
EXPAND
READVOL 21.     NEXT    100000.
EXPAND
READVOL 21.     NEXT    100000.
EXPAND
READVOL 21.     NEXT    100000.
EXPAND
QUIT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
EOFA

# run the program
cedric.e < cedric.inp
```

```
# a continuation of the Cray shell script
#
# dispose the cedric output file; CHANGE the filename here to yours
mswrite wdaft31 /ANDERSNB/TEST/output.mud

# save metacode file in a semi-permanent directory
# first, see if directory exists if so move file; if not, create then move file
# CHANGE 'andersnb' to your login id in the rest of this script
if (test -d /usr/tmp/andersnb) then
   mv gmeta /usr/tmp/andersnb/gmeta_test1
else
   mkdir /usr/tmp/andersnb
   mv gmeta /usr/tmp/andersnb/gmeta_test1
fi

# send color plotfile to color microfilm
sendtg /usr/tmp/andersnb/gmeta_test1 macr=viewercl qual=best
# send black and white plotfile to fiche
sendtg /usr/tmp/andersnb/gmeta_test1 macr=fiche titl=Cedric_Test_Run
# send black and white plotfile to Xerox printers on Mesa
plotmp /usr/tmp/andersnb/gmeta_test1
# generate accounting statistics
ja -cst
exit
```

Following are four simple setups to illustrate some of the basic aspects of the CEDRIC command structure. In all cases the above Unix shell script or one very similar to it must surround the CEDRIC commands. These setups are intended to convey some idea of how CEDRIC commands are structured, and how they are executed. Considerably more complicated setups are possible—some easily exceeding several hundreds of commands. This command structure can be viewed as a high-level (higher than Fortran) programming language that allows the user to construct a whole program, including subroutines (`DEFINE` blocks) that have been tailored to the user's needs.

+++++ Sample deck #1 +++++

Accessing a radar volume

The following deck accesses the first Cartesian radar volume found on the input tape assigned to unit 11 in the JCL. Statistics for the Z-planes of all the fields in the volume are also generated. This deck should be run by anyone using CEDRIC for the first time, since it is an excellent way to familiarize oneself with the required CRAY JCL without getting hung up on the details of the CEDRIC command set. The information following the `QUIT` command can be used as a positioning guide when supplying the parameterization required by CEDRIC commands, since any card images following the `QUIT` command are ignored.

```
READVOL 11.0    NEXT
COMMENT
    THIS COMMENT WILL BE SENT APPEAR IN THE OUTPUT LISTING.
    SO WILL THIS ONE !!!
COMEND
*
*   COMMANDS WITH AN (*) IN COLUMN 1 ARE ALSO TREATED AS COMMENTS
*
STATS           Z               ALL
QUIT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Sample deck #2 +++++

Accessing a radar volume and displaying data

The following deck accesses the first Cartesian radar volume found on the input tape assigned to unit `11.0` whose starting time is after `150000.` A spatial window is set to permit the optional viewing of only those Z-locations between `3.0` and `8.0` kilometers inclusive. A field window is also defined to permit designation of the two fields `DZNE` and `DMNE` as being PRIMARY so that they can be optionally referred to using the shorthand notation `PRI` in place of an actual field name. Statistics and `HISTO` displays are generated for `DZNE` and `DMNE` using this facility for all Z-planes. `CODED` and `DIGITAL` displays of both fields are produced for every second Z-plane within the optional spatial window. Finally, the spatial window is reset to permit the viewing of all Z and X locations, but to resrict Y-locations to the range `10.0` km to `40.0` km. `GRAPHICS` contour displays of the `DZNE` field for every third Y-plane within that window are subsequently produced.

```
COMMENT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
COMEND
READVOL 11.     NEXT    150000.         YES
WINDOW  DISTANCE                                3.0     8.0
FIELDSETPRI     DZNE    DMNE
STATS           Z               PRI
HISTO   PRINT   Z
        PRI     1.0     -120.0   70.0   10.0
END
CODED   PRINT   Z       2.0                                             WINDOW
        DZNE    1.0     0.0     5.0     0.      15.
        DMNE    1.0     -110.   5.0     6.      17.
END
DIGITAL PRINT   Z       2.0                                             WINDOW
        PRI     1.0     5.0
END
WINDOW  DISTANCE                10.0    40.0
GRAPHICS        Y       3.0     ++++ THIS IS TITLING INFORMATION ++++   WINDOW
        CONTOUR DZNE    CONT                    UNI     -10.    70.     10.
END
QUIT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Sample deck #3 +++++

Accessing a radar volume and editing the data

The following deck CREATEs a user defined Cartesian coordinate system into which all data from the radar volume called 8021809C is remapped upon transfer into CEDRIC. As in the previous examples, the input tape logical unit number is 11. The user can alter any of the (X,Y,Z) limits of the coordinate system as it appears on tape but may not modify the spacing. The DZNE field is filtered using the FILTER command; results are stored in a new field called DZFILT and statistics are generated for the new field. The FUNCTION command is used to produce the following new fields:

DZTHRSH contains the results of thresholding DZNE with a floor of 10.0 dBZ;

DZDIFF contains the difference between DZNE and DZFILT;

DZSUM contains the sum of DZNE and the absolute value of DZDIFF.

Finally, GRAPHICS contour and scatter plots are produced for every level, The fields DZSUM and DZTHRSH are DELETEd, and the edited volume is written to an output dataset called V12345 that has been assigned to fortran logical unit 35.

```
COMMENT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
COMEND
CREATE  -20.    30.     1.0     5.      70.     1.0     2.      12.     0.5
READVOL 11.     8021809C                YES
FILTER  DZFILT  DZNE    UNIFORM
STATS           Z               DZFILT
FUNCTIONNAME
        DZTHRSH PERM    FLOOR   DZNE            10.0
        DZDIFF  PERM    -       DZNE    DZFILT
        ABSDIF  TEMP    ABS     DZDIFF
        DZSUM   PERM    +       DZNE    ABSDIF
END
GRAPHICS        Z               SAMPLE ANALYSIS OF REFLECTIVITY FIELDS
        CONTOUR DZTHRSH BOTH    10101000        UNIFORM 10.     70.     10.
        CONTOUR DZDIFF  CONT                    UNIFORM -10.    10.     2.0
        SCATTER DZNE    -10.    70.     DZFILT  -10.    70.     1.0
END
DELETE  DZSUM   DZTHRSH
WRITVOL 35.     V12345  1809EDT BEG
QUIT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

+++++ Sample deck #4 +++++

Synthesis of (U,V,W) wind components from multiple Doppler radars

The following deck `CREATE`s a user defined Cartesian coordinate analysis space and `SYNTHES`izes radial velocity information from `6.0` independent Doppler radar volumes to produce the orthogonal components (U,V,W) of air motion. Differential advection is performed for the analysis time `181000` using a storm motion of `292.5` (deg) at `18.0` (M/S). Velocity and time fields are specified for each radar and reflectivity information is transferred to the output file from the first and second radars. These dBZ fields are renamed `DZCP2` and `DZCP3`, respectively. Uniform (W=`1.0`) weighting is used. The output of the `SYNTHES`is becomes the active edit file and a statistical display is generated for every field that was created. Each input volume must be assigned to a unique logical unit number. In this example units `41.0` through `46.0` were used. If more than one input volume is contained on a single dataset, transfers to a temporary unit may be performed prior to the `SYNTHES`is by using `READVOL` and `WRITVOL` commands. The `SYNTHES`is output volume `SYN1810` is written to an output dataset called `V45678` that has been assigned to fortran logical unit `51`.

```
COMMENT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
COMEND
CREATE  -30.    50.     1.0     0.0     80.     1.0     1.0     18.0    1.0
SYNTHES SYN1810 6.0     .05     100.    100.            181000. 292.5   18.0
        41.     1809                    YES     VFTH    DIF     TIME    YES
        INPUT   DZNE
        OUTPUT  DZCP2
        42.     1810                    YES     VFTH    DIF     TIME    YES
        INPUT   DZNE
        OUTPUT  DZCP3
        43.     1810                    YES     VFTH    DIF     TIME
        44.     1810                    YES     VETH    DIF     TIME
        45.     1810                    YES     VETH    DIF     TIME
        46.     1806                    YES     VETH    DIF     TIME
END
STATS           Z               ALL
WRITVOL 51.     V45678  SYN1810 BEG
QUIT
P1......P2......P3......P4......P5......P6......P7......P8......P9......P10.....
```

APPENDIX F
MULTIPLE-DOPPLER RADAR WIND SYNTHESIS IN CEDRIC

L. Jay Miller and William Anderson
Mesoscale and Microscale Meteorology Division
National Center for Atmospheric Research
Boulder, Colorado 80307

21 May 1991

1. Cartesian components of motion from radial velocities

The normal meteorological coordinate system consists of northward distance y, eastward distance x, and upward distance or height z above mean sea level of a curved earth. The velocity components corresponding to these coordinates are v, u, and w. All distances are in kilometers, and all velocities are in meters per second. The mapping to height above mean sea level is made in the interpolation (SPRINT) and analysis (CEDRIC) software packages, and it will not be explicitly listed here. Usually Doppler radars detect particles so there is an additional component of motion, fallspeed of particles in still air, to be considered. The normal convention of positive radial velocity away from the radar is used.

This mathematical formulation comes from the lead author's unpublished notes that have been accumulated over the years, and parallels the formulations presented by Armijo (1969), Bohne and Srivastava (1975), Ray et al. (1978) and Ray et al. (1980).

a. Mathematical formulation

The projection of particle motion $(\hat{u}, \hat{v}, \hat{w}, \hat{w}_t)$ along the Doppler radar radial direction is

$$\hat{v}_r = \hat{u}\ sin\ a\ cos\ e\ +\ \hat{v}\ cos\ a\ cos\ e\ +\ (\widehat{w + w_t})\ sin\ e, \tag{1}$$

where a and e are the azimuth and elevation angles of the beam. The azimuth angle is measured clockwise from true north (the positive y axis), and the elevation angle is measured upward from the horizontal plane through the radar. The circumflex indicates an average value within the radar sample volume defined by pulse length and beamwidth. Two or more radar measurements are usually combined along with the mass continuity equation to obtain the unknown components of air motion $(\hat{u}, \hat{v}, \hat{w})$. In the case of only two radar measurements additional information about the precipitation fallspeed, $\hat{w}_t$, is needed to obtain a solution for the four unknown components of motion. A similar procedure is followed with three or more radar measurements to separate the vertical air motion from the precipitation fallspeed.

It is more convenient to express the radial velocity in terms of Cartesian coordinates since solutions are obtained after interpolation of the radar measurements at locations in spherical coordinates (r, a, e). Further, if the radar echo (storm) is moving with components (U, V), and the radial velocity samples are taken at time $t + \Delta t$, the above equation is replaced with (adapted from Gal-Chen, 1982)

$$\frac{[\hat{v}_r r]_{t+\Delta t}}{[r]_t} = \left[\hat{u}\left(\frac{x - x_o + U\Delta t}{r}\right) + \hat{v}\left(\frac{y - y_o + V\Delta t}{r}\right) + \widehat{W}\left(\frac{z - z_o}{r}\right)\right]_t, \tag{2}$$

for a radar at (x_o, y_o, z_o) with slant range

$$r = \left[(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2\right]^{1/2}.$$

Radial velocities are first multiplied by slant ranges from the radar at the sample time $t + \Delta t$. This field is then advected at the storm motion to new locations, where it is divided by slant ranges at the synthesis time t. The coefficients of $\hat{u}$ and $\hat{v}$ on the right hand side of (2) have been modified to account for a change in radar pointing direction.

For M radars, the linear system of equations to be solved at time t is

$$\hat{u}a_m + \hat{v}b_m + \widehat{W}c_m = d_m \;;\; m = 1,\ 2,\ 3,\ \ldots, \tag{3}$$

where (a_m, b_m, c_m) are the coefficients and d_m is the term involving radial velocity on left hand side in (2), a set for each radar. In the method of least-squares, the error equation

$$Q = \Sigma E_m^2 = \Sigma(\hat{u}a_m + \hat{v}b_m + \widehat{W}c_m - d_m)^2 \tag{4}$$

is minimized with respect to the unknown quantities $(\hat{u},\ \hat{v}\ , \widehat{W})$. The resulting normal equations to be solved for three or two unknown quantities are:

$$\begin{aligned}
\hat{u}\Sigma a_m a_m + \hat{v}\Sigma a_m b_m + \widehat{W}\Sigma a_m c_m &= \Sigma a_m d_m \\
\hat{u}\Sigma b_m a_m + \hat{v}\Sigma b_m b_m + \widehat{W}\Sigma b_m c_m &= \Sigma b_m d_m \\
\hat{u}\Sigma c_m a_m + \hat{v}\Sigma c_m b_m + \widehat{W}\Sigma c_m c_m &= \Sigma c_m d_m,
\end{aligned} \tag{5}$$

or

$$\begin{aligned}
\hat{u}\Sigma a_m a_m + \hat{v}\Sigma a_m b_m &= \Sigma a_m d_m - \widehat{W}\Sigma a_m c_m \\
\hat{u}\Sigma a_m b_m + \hat{v}\Sigma b_m b_m &= \Sigma b_m d_m - \widehat{W}\Sigma b_m c_m.
\end{aligned} \tag{6}$$

For the linear system of three equations

$$\begin{aligned}
\hat{u}\ A_1 + \hat{v}\ B_1 + \widehat{W}C_1 &= D_1 \\
\hat{u}\ A_2 + \hat{v}\ B_2 + \widehat{W}C_2 &= D_2 \\
\hat{u}\ A_3 + \hat{v}\ B_3 + \widehat{W}C_3 &= D_3,
\end{aligned} \tag{7}$$

the solutions are:

$$\begin{aligned}
\hat{u} &= D^{-1}\big[D_1(B_2C_3 - B_3C_2) - D_2(B_1C_3 - B_3C_1) + D_3(B_1C_2 - B_2C_1)\big] \\
\hat{v} &= D^{-1}\big[D_1(A_3C_2 - A_2C_3) - D_2(A_3C_1 - A_1C_3) + D_3(A_2C_1 - A_1C_2)\big] \\
\hat{W} &= D^{-1}\big[D_1(A_2B_3 - A_3B_2) - D_2(A_1B_3 - A_3B_1) + D_3(A_1B_2 - A_2B_1)\big],
\end{aligned} \tag{8}$$

where the determinant of coefficients

$$D = A_1(B_2C_3 - B_3C_2) - A_2(B_1C_3 - B_3C_1) + A_3(B_1C_2 - B_2C_1).$$

The vertical component $\widehat{W}$ can be separated into air motion and fallspeed either by using $(\hat{u}, \hat{v})$ in the mass continuity equation to obtain $\hat{w}$ or by calculating fallspeed from radar reflectivity factor (or some other means).

For the system of two equations

$$\begin{aligned}
\hat{u}\ A_1 + \hat{v}\ B_1 &= D_1 - \widehat{W}C_1 \\
\hat{u}\ A_2 + \hat{v}\ B_2 &= D_2 - \widehat{W}C_2,
\end{aligned} \tag{9}$$

the solutions are

$$\begin{aligned}
\hat{u} &= \frac{D_1B_2 - D_2B_1}{D} + \widehat{W}\frac{B_1C_2 - B_2C_1}{D} = u' + \epsilon_u\widehat{W} \\
\hat{v} &= \frac{D_2A_1 - D_1A_2}{D} + \widehat{W}\frac{A_2C_1 - A_1C_2}{D} = v' + \epsilon_v\widehat{W},
\end{aligned} \tag{10}$$

where the determinant of coefficients

$$D = A_1B_2 - A_2B_1.$$

The quantities $\hat{u}$ and $\hat{v}$ depend not only on the radar measurements through (u', v') but also on an unknown $\widehat{W}$. The so-called over-determined, dual-Doppler approximations (e.g., Kessinger et al. 1987) use (u', v') as estimates of $(\hat{u}, \hat{v})$. Since ϵ_u and ϵ_v depend only on geometry, the impact of neglecting the vertical component in (10) can be assessed if the bounds of $\widehat{W}$ can be estimated.

It is clear from the preceding equations that the solutions for $(\hat{u}, \hat{v}, \widehat{W})$ are geometrically weighted sums of the interpolated radial velocities. Since the different radar radial velocity measurement errors are independent (the error in a particular radar measurement does not depend on other radar measurement errors even though the measurements themselves are related), the variance of the solutions given by (8) and (10) can be written as sums of radial velocity variance, weighted by the square of the geometric terms. From (8),

$$\begin{aligned} \sigma^2(\hat{u}) &= \Sigma g_{um}^2 \sigma^2(\hat{v}_m) \\ \sigma^2(\hat{v}) &= \Sigma g_{vm}^2 \sigma^2(\hat{v}_m) \\ \sigma^2(\widehat{W}) &= \Sigma g_{Wm}^2 \sigma^2(\hat{v}_m), \end{aligned} \tag{11}$$

which, if all radial velocity variances $\sigma^2(\hat{v}_r)$ are equal, can be rewritten as normalized variances

$$\begin{aligned} \sigma_N^2(\hat{u}) &= \frac{\sigma^2(\hat{u})}{\sigma^2(\hat{v}_r)} = \Sigma g_{um}^2 \\ \sigma_N^2(\hat{v}) &= \frac{\sigma^2(\hat{v})}{\sigma^2(\hat{v}_r)} = \Sigma g_{vm}^2 \\ \sigma_N^2(\widehat{W}) &= \frac{\sigma^2(\widehat{W})}{\sigma^2(\hat{v}_r)} = \Sigma g_{Wm}^2. \end{aligned} \tag{12}$$

Likewise, from (10)

$$\begin{aligned} \sigma_N^2(u') &= \frac{\sigma^2(u')}{\sigma^2(\hat{v}_r)} = \Sigma g_{um}^2 \\ \sigma_N^2(v') &= \frac{\sigma^2(v')}{\sigma^2(\hat{v}_r)} = \Sigma g_{vm}^2. \end{aligned} \tag{13}$$

These normalized variances can be used to estimate the impact of geometry on the transformation from non-orthogonal radial velocities to the orthogonal Cartesian components. This geometric transformation is such that errors in the solutions generally exceed errors in the measured radial velocities.

Once horizontal wind components are found, the mass continuity equation can be used to obtain the vertical component of air motion. There are two ways to proceed, the first uses

$$\frac{\partial(\rho w)}{\partial z} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0,$$

so that

$$\int_{z_k}^{z_{k+1}} \frac{\partial(\rho w)}{\partial z} dz = -\int_{z_k}^{z_{k+1}} \rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) dz$$

or, in finite difference form,

$$(\rho w)_c = (\rho w)_p - \delta \Delta z \overline{\left[\rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]_{p-c}}, \tag{14}$$

where

$$\delta = \begin{cases} +1, & \text{for upward integration} \\ -1, & \text{for downward integration.} \end{cases}$$

The overbar represents an average of divergence values from the previous (p) and current (c) levels. The boundary condition $(\rho w)_b$ must be specified at every (x, y, z_b) at the bottom of the domain if integrating upward or at the top of the domain if integrating downward. This can be done by setting the boundary value to a constant, by using a fraction of the first level divergence times a height increment, or by constructing a field using options such as found in the FUNCTION command. Equation (14) can also be integrated using a variational scheme where both upper and lower boundary conditions are specified within each integration column. Vertical motions rather than density-weighted values are output, where the user specifies the density weighting (an exponential form is provided, but another form can be constructed with the FUNCTION command).

An alternative path is available that is more mathematically rigorous when (u, v) come from the two-equation solution. The first step is to correct (u', v') for precipitation fallspeed using (10),

$$\begin{aligned} u_2 &= u' + \epsilon_u \hat{w}_t \\ v_2 &= v' + \epsilon_v \hat{w}_t, \end{aligned}$$

then substitute

$$\begin{aligned} u &= u_2 + \epsilon_u w \\ v &= v_2 + \epsilon_v w \end{aligned}$$

into the mass continuity equation and integrate. The proper form of the mass continuity equation becomes

$$\frac{\partial(\rho w)}{\partial z} + \frac{\partial(\rho u_2)}{\partial x} + \frac{\partial(\rho v_2)}{\partial y} + \frac{\partial(\rho \epsilon_u w)}{\partial x} + \frac{\partial(\rho \epsilon_v w)}{\partial y} = 0.$$

This partial differential equation can no longer be solved as done in (14) since the desired vertical component appears in the two terms involving horizontal derivatives. These terms cause the solutions for w at adjacent (x, y) grid points to be coupled. In finite difference form, the mass continuity equation becomes

$$\underbrace{(\rho w)_c}_{A} = \underbrace{(\rho w)_p}_{B} \underbrace{- \delta \Delta z \overline{\left[\rho\left(\frac{\partial u_2}{\partial x} + \frac{\partial v_2}{\partial y}\right)\right]}_{p-c}}_{C} \underbrace{- \delta \Delta z \overline{\left[\frac{\partial(\rho \epsilon_u w)}{\partial x} + \frac{\partial(\rho \epsilon_v w)}{\partial y}\right]}_{p-c}}_{D}. \tag{15a}$$

Term D consists of two parts:

$$\underbrace{-\frac{\delta \Delta z}{2}\left[\frac{\partial(\rho \epsilon_u w)}{\partial x} + \frac{\partial(\rho \epsilon_v w)}{\partial y}\right]_p}_{D1} \underbrace{-\frac{\delta \Delta z}{2}\left[\frac{\partial(\rho \epsilon_u w)}{\partial x} + \frac{\partial(\rho \epsilon_v w)}{\partial y}\right]_c}_{D2}, \tag{15b}$$

where $D1$ is a function of horizontal derivatives at the previous level and $D2$ is a function of horizontal derivatives at the current level of integration.

To understand this integration scheme, start at a level where boundary conditions have first been applied. The horizontal divergence in (15a) is adjusted in the manner dictated by term $D1$, and $(\rho w)_c$ is estimated by integrating one level in the vertical direction. These first estimates of vertical motion at the current level are then multiplied by the (ϵ_u, ϵ_v) geometric factors and differentiated according to $D2$ in (15b) to readjust the layer convergence, and (15a) is reintegrated. Since only A and $D2$ change, for the nth iteration

$$A_n = B + C + D1 + D2_{n-1},$$

which proceeds from the previous to current levels until A in (15a) stops changing. The change in A is calculated at each grid point, and these differences are globally averaged for comparison with ϵ_o, a parameter set by the user:

$$\mid \overline{(\rho w)_{c,n} - (\rho w)_{c,n-1}} \mid \leq \epsilon_o.$$

Once this condition is satisfied, iterative integration is done from the current to the next level and so on until all levels have been integrated.

b. Synthesis tests

The synthesis in CEDRIC outputs several fields, depending on whether the three-equation or the two-equation solution is selected:

U: component of motion in the x direction, either $\hat{u}$ in (8) or u' in (10),

V: component of motion in the y direction, either $\hat{v}$ in (8) or v' in (10),

W: upward component of motion, $\widehat{W}$ in (8),

USTD: normalized standard deviation, square root of normalized u-variance in (12) or (13),

VSTD: normalized standard deviation, square root of normalized v-variance in (12) or (13),

WSTD: normalized standard deviation, square root of normalized W-variance in (12), available only when three-equation solution is selected,

EWU: geometric factor that multiplies vertical component $\widehat{W}$ in (10), available only when two-equation solution is selected,

EWV: geometric factor that multiplies vertical component $\widehat{W}$ in (10), available only when two-equation solution is selected.

The user specifies input parameters $DTEST1, DTEST2, DTEST3$ against which the above geometric parameters are tested. Selection of the two- or three- equation solution along with these test values determines the synthesis outcome at each (x, y, z) grid point. If only one radial velocity is present, no solution exists and all output fields are set to a bad data flag.

Two-equation solution (the number of radars, M $\geq$2):

if the magnitudes of (EWU, EWV) are both less than $DTEST1$,

and $(USTD, VSTD)$ are both less than $DTEST2$,

the output (U, V) are given by (u', v') in (10).

Three-equation solution is selected (the number of radars, M $\geq$3):

if $(USTD, VSTD)$ are both less than $DTEST2$,

the output (U, V) are given by $(\hat{u}, \hat{v})$ in (8),

and if $WSTD$ is less than $DTEST3$,

the output W is given by $\widehat{W}$ in (8).

If only two radial velocities are present at the grid point, the procedure follows the one for a two-equation solution.

2. Coplane components of motion from radial velocities

Lhermitte and Miller (1972) first introduced COPLAN radar scanning where measurements are taken in a cylindrical coordinate system that is more natural for two radars (Fig. 1). One axis is along the baseline joining the two radars, another is perpendicular to the baseline, and the third is the coplane (dihedral) angle associated with a series of planes defined by the two radar beams and the horizontal plane passing through the radars. This coordinate system simplifies the mathematical formulation, and it is the same one used by Armijo (1969) in his transformation that was needed to solve the mass continuity equation for vertical motion. Miller and Strauch (1974) expanded on the coplane coordinate concept, including a detailed analysis of the errors involved in the synthesis. Doviak et al. (1976) also dealt with error analysis, and they included errors associated with the integration of the mass continuity equation. A series of papers by French scientists (Testud and Chong; Chong et al.; and Chong and Testud, 1983) revisited all aspects of coplane analysis, including interpolation, synthesis, and integration of the mass continuity equation.

a. Mathematical formulation

When the original radar sampling process is done in coplanes formed by the beams from two radars, a procedure similar to the one for the two-equation solution is followed. If only two radar measurements are available, rewrite (1) as

$$
\begin{aligned}
\hat{u}\sin a_1' + \hat{v}\cos a_1' &= \frac{\hat{v}_1}{\cos e_1} - \widehat{W}\tan e_1 \\
\hat{u}\sin a_2' + \hat{v}\cos a_2' &= \frac{\hat{v}_2}{\cos e_2} - \widehat{W}\tan e_2,
\end{aligned}
\tag{16}
$$

where the angle $a' = a - a_o$ is measured from the baseline (at azimuth angle a_o) that joins the two radars. Equation (16) has the solution

$$
\begin{aligned}
\hat{u} &= D^{-1}\left[\frac{\hat{v}_1\cos a_2'}{\cos e_1} - \frac{\hat{v}_2\cos a_1'}{\cos e_2}\right] - \widehat{W}\tan\alpha \\
\hat{v} &= D^{-1}\left[\frac{\hat{v}_2\sin a_1'}{\cos e_2} - \frac{\hat{v}_1\sin a_2'}{\cos e_1}\right],
\end{aligned}
\tag{17}
$$

where $D = \sin(a_1' - a_2') = \sin(a_1 - a_2)$, and the coplane relations (Lhermitte and Miller, 1970)

$$
\tan\alpha = \frac{\tan e_i}{\sin a_i'};\ i = 1,2 \tag{18}
$$

have been used. The quantity α is the dihedral angle between the coplane defined by the two intersecting radar beams and the horizontal plane passing through the two radar locations. The radar beam is steered in the elevation angle direction as it is rotated in azimuth for a fixed coplane angle according to (18). Measurements are taken at range-azimuth locations in a series of these planes, with α increasing upwards from the horizontal plane of the two radars.

An alternative to (17) that will uncouple the vertical component from $\hat{u}$ is to introduce coplane coordinates

$$
\begin{aligned}
x_c &= (x^2 + z^2)^{1/2} \\
y_c &= y \\
\alpha &= Tan^{-1}(z/x),
\end{aligned}
$$

and solve for components of the motion within coplanes:

$$
\begin{aligned}
\hat{u}_c(x_c, y_c) &= \frac{r_1\hat{v}_1(y_c - y_{c1}) - r_2\hat{v}_2(y_c - y_{c2})}{2dx_c} \\
\hat{v}_c(x_c, y_c) &= \frac{r_2\hat{v}_2 - r_1\hat{v}_1}{2d},
\end{aligned}
\tag{19}
$$

where $2d$ is the distance between the two radars located at $(0, y_{c1})$ and $(0, y_{c2})$. The coplane coordinates x_c and y_c are measured perpendicular and parallel to the radar baseline. Since the component of motion perpendicular to the radar beam is not measured by the radar, the component normal to the coplanes does not appear in (19). In coplanes, distances from the radars are

$$
\begin{aligned}
r_1 &= \left[x_c^2 + (y_c - y_{c1})^2\right]^{1/2} \\
r_2 &= \left[x_c^2 + (y_c + y_{c2})^2\right]^{1/2}.
\end{aligned}
$$

Normalized variances within each coplane are similar in form to the ones for Cartesian components in (13). The two-equation Cartesian solution (in a relative coordinate system with y along the two-radar baseline) and the coplane solution give identical results in the horizontal plane ($\alpha = 0°$).

After $\hat{u}_c$ is corrected for fallspeed, the coplane components of air motion are:

$$
\begin{aligned}
u_c' &= \hat{u}_c - \hat{w}_t\sin\alpha \\
v_c' &= \hat{v}_c.
\end{aligned}
$$

These components are integrated in the mass continuity equation written in the form

$$\frac{1}{x_c}\frac{\partial(\rho w_c')}{\partial\alpha} + \frac{1}{x_c}\frac{\partial}{\partial x_c}(\rho x_c u_c') + \frac{\partial}{\partial y_c}(\rho v_c') = 0 \tag{20}$$

to obtain the component normal to the coplanes. A finite difference form of (20) similar to (14) is used, with an exponential density weighting provided. Once these coplane components are calculated, horizontal wind components are obtained:

$$\begin{aligned}\hat{u} &= u_c' \cos\alpha - w_c' \sin\alpha \\ \hat{v} &= v_c' \\ \hat{w} &= u_c' \sin\alpha + w_c' \cos\alpha.\end{aligned} \tag{21}$$

The velocity transformation in (21) is done when the winds at coplane coordinates are interpolated to Cartesian coordinates with the REMAP command.

b. Synthesis tests

For coplane synthesis, CEDRIC outputs a subset of fields:

U: component of motion perpendicular to the baseline, $\hat{u}_c$ in (19)

V: component of motion parallel to the baseline, $\hat{v}_c$ in (19)

USTD: normalized standard deviation, similar to the square root of normalized u-variance in (13),

VSTD: normalized standard deviation, similar to the square root of normalized v-variance in (13),

The user specifies input parameters $DTEST2$ and $DTEST3$ against which the above geometric parameters are tested. Selection of coplane coordinate option (housekeeping words are tested or the COORD command is used) along with these test values determines the synthesis outcome at each (x_c, y_c, c) grid point. If only one radial velocity is present, no solution exists and all output fields are set to a bad data flag.

Two-radar coplane solution (the number of radars, M = 2):

if $(USTD, VSTD)$ are both less than $DTEST2$,

the output (U, V) are given by (u_c, v_c) in (19).

(EWU, EWV) are zero in coplane coordinates so they are not output.

The following steps are taken for multiple-Doppler radar wind synthesis:

1. Interpolate the radar measurements using the SPRINT software package. If measurements were originally taken in the coplane coordinate system, interpolation within these coplanes is selected. If measurements were taken in the normal radar spherical coordinate system, interpolation to Cartesian coordinates should be done. There is little advantage in interpolating to coplanes if the data were not taken in this coordinate system. The SPRINT package has no provision for such interpolation; however, the REORDER package of ATD/RDP does.

2. Select the appropriate coordinate system synthesis, integration and interpolation path in CEDRIC. This is done automatically if the appropriate housekeeping words are correctly set. Otherwise, the user must invoke the COORD command to force the correct choice of mathematical formulation.

3. Specify the necessary parameters in the SYNTHES command in CEDRIC. The three-equation solution fields come from (8) and (12). The Cartesian synthesis allows for an over-determined, two-equation solution with the various fields given by (10) and (13). For coplane synthesis at coplane coordinates, only two radars are allowed, and the solutions come from (19).

4. Invoke the CONVERGE command to compute horizontal or coplane convergence, the negative of divergence of the horizontal winds in (8) or (10) or the coplane winds in (19).

5. Specify the necessary parameters in the INTEGR command for Cartesian (14) or coplane (20) integration. If the user wishes to use the iterative integration of (15), the MASS2 command is used.

References:

Armijo, L., 1969: A theory for the determination of wind and precipitation velocities with Doppler radars. *J. Atmos. Sci.*, **26**, 570–575.

Bohne, A.R., and R.C. Srivastava, 1975: Random errors in wind and precipitation fall speed measurement by a triple Doppler radar system. Univ. of Chicago, Tech. Report No. 37, 50 pp.

Chong, M. and J. Testud, 1983: Three-dimensional wind field analysis from dual-Doppler radar data. Part III: The boundary condition: An optimum determination based on a variational concept. *J. Climate and Appl. Meteor.*, **22**, 1227-1241.

Chong, M., J. Testud and F. Roux, 1983: Three-dimensional wind field analysis from dual-Doppler radar data. Part II: Minimizing the error due to temporal variation. *J. Climate and Appl. Meteor.*, **22**, 1216-1226.

Doviak, R.J., P.S. Ray, R.G. Strauch, and L.J. Miller, 1976: Error estimation in wind fields derived from dual-Doppler radar measurements. *J. Appl. Meteor.*, **15**, 868-878.

Gal-Chen, T., 1982: Errors in fixed and moving frames of references: Applications for conventional and Doppler radar analysis. *J. Atmos. Sci.*, **39**, 2279-2300.

Kessinger, C.J., P.S. Ray, and C.E. Hane, 1987: The Oklahoma squall line of 19 May 1977. Part II: A multiple Doppler analysis of convective and stratiform structure. *J. Atmos. Sci.*, **44**, 2840-2864.

Lhermitte, R.M., and L.J. Miller, 1970: Doppler radar methodology for the observation of convective storms. *Preprints, 14 Conf. Radar Meteorology*, 133-138.

Miller, L.J., and R.G. Strauch, 1974: A dual Doppler radar method for the determination of wind velocities within precipitating weather systems. *Remote Sens. Environ.*, **3**, 219-235.

Ray, P.S., K.K. Wagner, K.W. Johnson, J.J. Stephens, W.C.Bumgarner, and E.A. Mueller, 1978: Triple-Doppler observations of a convective storm. *J. Appl. Meteor.*, **17**, 1201-1212.

Ray, P.S., C.L. Ziegler, W. Bumgarner, and R.J. Serafin, 1980: Single- and multiple-Doppler radar observations of tornadic storms. *Mon. Wea. Rev.*, **108**, 1607-1625.

Testud, J., and M. Chong, 1983: Three-dimensional wind field analysis from dual-Doppler radar data. Part I: Filtering, interpolating and differentiating the raw data. *J. Climate and Appl. Meteor.*, **22**, 1204-1215.

APPENDIX G

COORDINATE SYSTEMS THAT CAN BE USED IN CEDRIC

The most common coordinate system that data will be in for use in CEDRIC is a three dimensional Cartesian coordinate system. A staggered grid is not allowed. There are other specialized coordinate systems that can be used, however.

One of these is a coplane coordinate system. This is only a brief discussion of the coplane coordinate system. Users unfamiliar with the coplane coordinate system and its associated mathematics should examine Miller and Strauch (1974) and Appendix F in the CEDRIC documentation.

As a radar antenna rotates in azimuth, the elevation angle is continuously changed to keep the beam in a flat plane passing through the baseline joining two radars. This plane contains the intersecting beams from the two radars and the baseline. In this scanning scheme, measurements are taken at range-azimuth locations within each plane nearly simultaneously by the two radars, and then the two antennaes step to the next tilted coplane. The coplane angle is the angle formed by the tilted coplane and the horizonal plane through the radar positions, and it is measured upward from the horizontal tangent plane.

The gridded coplane coordinate system consists of three orthogonal directions: if you look in a direction from one radar toward the other (along the baseline between the radars) and with the region of intereset to your right then the positive X-direction is perpendicular and outward from the baseline toward the region of interest, the positive Y-direction is parallel to the baseline in the direction you are looking, and C remains the coplane angle. Within each coplane, X and Y constitute a two-dimensional Cartesian coordinate system. Data in this coordinate system can be displayed and analyzed as in a Cartesian system. Beware of displays where a dimension other than C is held fixed, however. No stretching is done to account for the curved coordinate system.

Another kind of coordinate system usable in CEDRIC is a PPI-like coordinate system where the levels are constant elevation surfaces instead of constant height surfaces. Graphical displays and some statistical output can be generated for data in this coordinate system; many analysis commands will not work for this coordinate system, however. For the most part, only displays should be generated.