Internal Report No. 40

DATA FORMATS FOR OGO-6 CERENKOV DATA PROCESSING

by James W. Brown

6 Oct. 1972

i. ABSTRACT

This report describes tape formats and data content for all tapes labelled CERAnn, CERPnn, CERPEn, CERXnn, CRACnn, and a small number of additional tapes with similar labels. Tapes labelled CERnnn are described in SRL Internal Report No. 29.

ii. CONTENTS:

DATA FORMATS FOR OGO-6 CERENKOV DATA PROCESSING

I. CERA tapes; PINTERP program

PINTERP is a S/370 program which calculates vertical cutoff rigidities from a modified Shea and Smart 2° x 2° grid of rigidities calculated for an altitude of 20 km. A 2° x 2° grid of Jensen and Cain 1960 L-values is used to map these rigidities to satellite altitude. Input to the program is a "final edit" CER tape (cf. SRL Int. Rept. No. 29), and the cutoff and L-value grids. Output is a CERA tape containing the same data as the input tape, plus interpolated cutoff rigidities. The format is the same as a CER tape, except that an extra A-record is inserted immediately preceding each existing A-record (except for the first A-record of each acquisition). This extra A-record, which may then be called a "P-record," is distinguished from the real A-records by having an orbit number of ZERO. Words 2 thru 6, instead of containing orbit data, contain instead five integer rigidities as follows:

Let T_1 be the universal time (UT) corresponding to the previous A-record; let T_2 be the UT for the next (immediately following) A-record. Let $X(t)$ be the set of orbit parameters corresponding to time t, calculated from $X(T_1)$ and $X(T_2)$ by linear interpolation, where $T_1 < t < T_2$. Then let $P(X(t))$ be the rigidity calculated for point X by linear interpolation in the 2° x 2° grids, expressed in units of 0.01 GV, as an integer. Then the five rigidities in each P-record are:

$$P_1 = P(X(T_1))$$

$$P_2 = P(X((3T_1+T_2)/4))$$

$$P_3 = P(X((T_1+T_2)/2))$$

$$P_4 = P(X((T_1+3T_2)/4))$$

$$P_5 = P(X(T_2))$$

The output tapes each contain one calendar month of data in chronological order: CERAØ1 contains data from June 1969, and CERA15 contains August 1970. CERBØ1 is a specially edited version of CERAØ1, with acquisitions containing MC114(2) scaler malfunction deleted.

II. CERP tapes; CRBINS program

CRBINS is a PDP-11 program which assigns an interpolated rigidity to each analyzed event. This rigidity is calculated by linear interpolation between two of the five rigidity values available in each P-record of a CERA tape. An interpolated longitude is also assigned, and a flag indicating whether the event was observed in a trapped particle region. The latter calculation is done as follows:

1) For each interval between A-records (usually one minute), the following are accumulated:
   x = number of analyzed events, y = number of D2'D3'$\overline{D4}$' coincidences recorded by MC114(2), z = number of non-errored readouts of MC114(2). Records with any error flags set are ignored.
2) If $y > 32767$ then the entire interval is considered to represent trapped particles.
3) If $x/y \geq r$ and $3y \leq z$, then the interval is considered "non-trapped." Here r is a parameter between ~ 0.5 and ~ 0.9, whose value is determined by trial and error. Currently it is set at 0.75.
4) If $10y \leq z$, then the interval is non-trapped.
5) Otherwise ($10y > z$ and $x/y < r$, or $3y > z$) the interval is considered trapped.
6) If the interval is considered trapped, it is further classified as "inner zone," "outer zone," or "unidentified," on the basis of the cutoff rigidities spanned.

Each event is written as an 8-byte logical record as follows:

BYTE	CONTENTS
1	0 = non-trapped, 1 = inner zone, 2 = outer zone, 3 = unidentified, -1 = 255 = label*
2	D1' channel number, or -1 = 255 if label*
3	D2' channel number, or LFLAGS if label*
4	D3' channel number,
5-6*	Rigidity, in units of 0.01 GV (integer)
7-8*	Longitude, $0 \leq$ long. < 360 (integer)

* For each input label record, an output record is written with -1 in bytes 1-2, LFLAGS in bytes 3-4, and the run name in bytes 5-8. The low order bits of LFLAGS are in byte 3 (PDP-11 order) and must be swapped with byte 4 if read on S/370. The run name is in ASCII. A fill record consists of 8 bytes of -1 (255; i.e. all bits on). These 8-byte logical records are packed into 2048-byte physical records on the CERP tapes.

CERP tapes are numbered the same as CERA tapes; e.g. CERPØ1 contains June 1969 data, etc. CERP tapes are 7-track, 800 bpi, written in PDP-11 "coredump mode," i.e., 2 tape characters per 8-bit byte.

III. CERPE tapes; CERPED program

CERPE tapes are condensed versions of the CERP tapes discussed above. They are produced by CERPED on the PDP-11. This program selects events with a particular trapping flag (usually zero), and a rigidity lying between limits specified at run time. These events are written onto CERPE tapes as four-byte logical records as follows:

BYTE	CONTENTS
1	Rigidity, in units of 0.08 GV
2	D1' channel number
3	D2' channel number
4	D3' channel number

Four bytes all zero indicate a fill record. These are inserted to fill out the 4096-byte physical records at each input EOF.

CERPED can optionally accept a CERPE tape as input, and produce a subset of the input events as its output. CERPE tapes may be 7-track or 9-track, and their contents depend on parameters specified for each run. A log is kept in one of the program binders indicating the contents of each tape file.

IV. CERX tapes; CTWOD, C2DP, and C2DADD programs

CTWOD and C2DP are PDP-11 programs (not to be confused with the S/370 version of CTWOD, which has a similar though not identical function) which read CER or CERA tapes (CTWOD), or CERP or CERPE tapes (C2DP), and produce two-dimensional accumulations of analyzed events. CTWOD processes all analyzed events, while C2DP selects events on the basis of trapping flags and rigidity. Each "axis" of the 2-D output may be any of: D1', D2', D3', or MIN(D1',D2'). The programs run multiple passes through the input tape, accumulating 4096 channel combinations on each pass. These combinations are usually one of two types: (1) 16 passes each handling a 64-by-64 channel array; (2) 4 passes each handling a 128-by-128 channel array with a factor-of-2 decrease in resolution along each axis. In both cases the output arrays are 64-by-64 arrays of 16-bit integers containing the number of analyzed events in each channel-pair. The resulting array may be listed on the teletype and/or written as an 8192-byte record on a CERX tape. Each file on a CERX tape represents one run of either program, and contains one record for each pass (up to a maximum of 16 passes per run).

C2DADD takes one or more records from CERX tape(s), adds or subtracts into a 4096-word accumulating array with optional factor-of-2 compression along each axis, and optionally writes the resulting array on the teletype and/or another CERX tape. It is used for combining the results of several C2DP or CTWOD runs, or compressing their output (factor-of-2) and thus combining four input records into a single output record.

V. CRAC tapes; CRACCM program

This is the fundamental (PDP-11) program for obtaining spectra. The input is a CERA tape, which is processed for removal of trapped particles in a manner identical to the processing in CRBINS (see above). The output is a CRAC tape which contains one 6148-byte record for each input run (i.e., CERENKOV program run).

The CRAC tapes are in PDP-11 format (i.e. low order byte first), and may be 7-track or 9-track. The records contain:

BYTE	CONTENTS
1-4	Run name (in ASCII)
5-2052	NREAD
2053-4100	NEVNT
4101-6148	NCREC

The last three items are 4-by-256 arrays of 16-bit words containing, respectively, the number of readouts of MC114(2), the number of D2'D3'$\overline{D4}$' coincidences counted by MC114(2), and the number of analyzed events. Each array is of the form NREAD(I,J), where I is one plus the trapping flag (cf. para. II) (I = 1,2,3,4 for non-trapped, inner zone, outer zone, unidentified), and J is one plus the rigidity expressed in units of 0.08 GV. The array elements are stored in "FORTRAN order," i.e., NREAD (1,1), NREAD(2,1), NREAD(3,1), NREAD(4,1), NREAD(1,2), etc. Accumulations cover only the run named in the record. Arrays are zeroed before each run.