(9) NCLSZM: No. of clusters in the Z < 0 end cap

(10) NCLSZP: " in the Z > 0 end cap

(11) ETOT : total energy including charged track contribution

(12) ETOTB : " in the barrel part

(13) ETOTZM: " in the Z < 0 end cap

(14) ETOTZP: " in the Z > 0 end cap

(15)* NNEUT : No. of photons

(16)* EGTOT : total energy of photons

(17)* EGTOTB: " in the barrel part

(18)* EGTOTM: " in the Z < 0 E.C.

(18)* EGTOTP: " in the Z > 0 E.C.

(20) IER : The flag for any error in the course of cluster analysis. (=0 if no error)

(21) ISTEP : Stage of the analysis step
= 1 for step 1
= 2 if step 2 is finished

(22) ICORR : = 1 if detailed energy correction is done

(23)
not used kept for future use to put other flags.
(24)

(25) NWPCL : No. of words used per cluster for the cluster information (= 15, at present)

Cluster Information

/ Cluster Map /

(26) MAP(1): Start position of cluster 1 in the shuffled ADC data

(27) MAP(2):

.
.
.

Fit Parameters

- circle fit in xy-plane (type = 1)

19 P1 = curvature = R^{-1} $[mm^{-1}]$

20 P2 = D_o - R [mm] D_o = distance (coordinate origin-centre of circle)

21 P3 = angle of direction (coordinate origin-centre of circle)
ε $[-\pi, +\pi]$

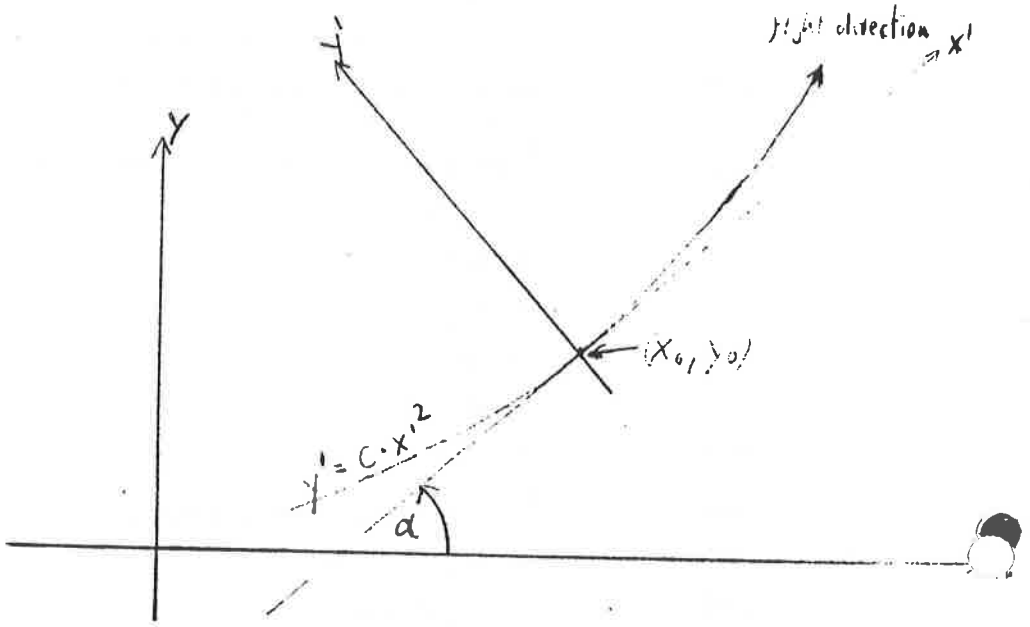
- parabola fit in xy-plane (type = 2): $y' = c \cdot x'^2$

19 P1 = α $\varepsilon[-\pi, +\pi]$

20 P2 = x_o

21 P3 = y_o

22 P4 = c



α = angle between y-axis and y'-axis

- straight line fit in rz-plane (type = 1)

$z = P1 \cdot r + P2$ $\qquad r = \sqrt{x^2 + y^2}$

30 P1
31 P2

- data for a track:

(1) : track number

(2) : program identifier. Using IBM notation this word is coded as follows:

Bit 31 set → fast pattern recognition
Bit 30 set → medium " "
Bits 30 and 31 set → slow pattern recognition
Bit 29 set → TP momentum fit
Bit 28 set → manually edited

For PATR bank 12 (Monte Carlo created bank)
hdata (3) = no. of track generating four vectors stored in bank VECT 0
hdata (4) = type of particle as in VECT bank

(3) : date at which pattern recognition run

(4) : type (e.g.: conversion point, decay point, ...)
(5) : x
(6) : y
(7) : z
(8) : dx
(9) : dy (direction of flight) (unit vector)
(10) : dz

} of 1. measured point of the track

(11)
.
. : type, x,y,z dx, dy, dz of last measured point of the track
.
.
(17)

(18) : type (e.g.: parabola, circle)
(19)
(20)
(21) : parameters
(22)
(23) : $\sqrt{\chi^2/D.O.F.}$ (mm)
(24) : No. of hits used in the fit
(25) : curvature
(26) : δ(curvature)
(27) : curvature at 1. measured point
(28) : " " last " "

1 circle
2 parab.

} of fit in the x-y plane

signed quantity!

(29) : type
(30) :
(31) : parameters
(32) : $\sqrt{\chi^2/D.O.F.}$ (mm)
(33) : No. of hits used in the fit

} of fit in the r-z plane

- data for a track:

(1) : track number
(2) : identifier of program
(3) : date of production

(4) : type (e.g.: conversion point, decay point, ...)
(5) : x
(6) : y
(7) : z
(8) : dx
(9) : dy
(10) : dz

(8)–(10): (direction of flight) (unit vector)

(4)–(10): of 1. measured point of the track

(11)–(17): : type, x, y, z dx, dy, dz of last measured point of the track

(18) : type (e.g.: parabola, circle)
(19)–(22): : parameters
(23) : χ^2
(24) : No. of hits used in the fit
(25) : curvature
(26) : δ(curvature)
(27) : curvature at 1. measured point
(28) : " " last " "

(18)–(28): of fit in the x-y plane

(29) : type
(30)–(31): : parameters
(32) : χ^2
(33) : No. of hits used in the fit

(29)–(33): of fit in the r-z plane

(34)–(39): : cell numbers that contain hits of the track (in the same order as passed by the track)

(40) : pointer to corresponding lead glass cluster
(41) : " " " μ-chamber hits
(42) : " " " track bank in TP-bank
(43) : " " " TOF-bank
(44)–(47): free for other pointers

(48) : error code (= 0 if everything is ok)

JBCRE : The creation of a BOS-bank can be done by the

```
CALL JBCRE (IND, NA, NR, NW, IERR)
```

NA = name of the bank
NR = number of the bank
NW = length of the bank (4 byte words)
IND = pointer to first data word -1
IERR = 0 if the bank has been created successfully
= 1 if bank with same name and number already existing
= 2 if no more space available
= 3 if name of the bank is unknown

JBCR0 : The creation of a bank that is initialized to zero can be done by the

```
CALL JBCR0 (IND, NA, NR, NW, IERR)
```

JBCRA : The creation of a bank that is filled from the array AR can be done by the

```
CALL JBCRA (IND, NA, NR, NW, IERR, AR(1) )
```

JBDLS : A single bank can be deleted by the

```
CALL JBDLS (NA, NR, IERR)
```

IERR = 0 if bank has been deleted
= 1 if bank of the requested number does not exist
= 3 if NA is an unknown bank name

The bank will be deleted and all pointers readjusted.

JBDLM : All banks of a name NA can be deleted by the

```
CALL JBDLM (NA, IERR)
```

IERR = 0 if banks have been deleted
= 3 if NA is an unknown name

b) If you have more than 1 library to link with the L step must contain

```
//L.SYSLIB  DD  DSN = USER.XM65.JADEBOS,DISP=SHR
//          DD
//          DD
//          DD
//          DD  DSN = OTHER.LIBRARY,DISP=SOMETHING
//          DD  DSN = ANOTHER.LIBRARY,DISP=SOMETHING
etc.
```

The automatic call structure acts similarly to NEWLIB and ensures that only the required parts of the BOS library are loaded.

Notes

1. Bank names

Bank names have 4 alphanumeric characters. The last character must not be a '2' as this designates a low priority bank and these are no longer used.

2. Optimisation

The user is advised to read section 12 of the BOS report. Since BOS is a sophisticated program a price has to be paid in the time taken to execute various functions. Section 12 tells of one or two ways in which execution may be speeded up.

In addition it is useful to note that the only action performed by SUBROUTINE BSTR is to call an assembler routine to copy the words from the source array to the destination work space. To avoid this double-call the user may use the following code instead of CALL BSTR (IND,JW,NW).

```
IBM's:-     CALL UCOPY2 (JW,IW(IND+1),NW)
NORD :-     DOUBLE INTEGER IW
            COMMON /BCS / IW (1000)
            INTEGER HW (2000)
            EQUIVALENCE (HW(1),IW(1))
                        .
                        .
                        .
                        .
                        .
            NWH = 2 * NW
            NSTART = 2*(IND+1)-1
            CALL ZCOPY (JW,HW(NSTART),NWH)
                        .
                        .
                        .
```

This will save time when frequent storage of banks occurs.

3. If at any time the BOS source is recompiled remember that PARM.FORT = XL must be specified in the compilation as BOS uses the FORTRAN H inline logical functions.

4. It is assumed that at some later time a subroutine will be added to the package to allow updating of the fixed-pointer table being used by JADE (see JADE Note 24).

Routine Call	Purpose	Notes
CALL BMLT (N*,LIST*)	Define a set of N bank names	LIST must be DOUBLE INTEGER for the NORD 10.
CALL BPRM	Print a set of banks	Banks defined in BMLT are printed.
CALL BPRS (NA*,NR*)	Print contents of bank name NA number NR	NA must be DOUBLE INTEGER on NORD 10.
CALL BSAT (N*, LIST*) CALL BSAW (N*, LIST*)	Two routines to update the special list defined on BMLT.	
CALL BSLT CALL BSLW	More bank definition routines (with markers).	See BOS Note.
CALL BSTA	Printing of final statistics at end of program.	
CALL BSTR (IND*,JW*,NW*)	Store NW words from array JW into work array IW starting at location IND.	JW must be DOUBLE INTEGER on the NORD.
CALL BWRITE (IUN*)	Output a set of banks previously defined to unit IUN.	Garbage collection is done if necessary.

FORTRAN Implementation

The BOS package is entirely FORTRAN, although one or two assembler subroutines are used internally. Anyone wanting to use BOS must have in his main program the following statements:

```
COMMON/BCS/IW (NSPACE)
REAL RW (NSPACE)
EQUIVALENCE (IW(1), RW(1))
```

where NSPACE = an integer constant defining the length of the storage work space IW.

Note that on the NORD 10 the statement

```
DOUBLE INTEGER IW (NSPACE)
```

must also be included to ensure that storage space is in 4 byte words as on the IBM's.

On the next pages is an alphabetical list of the FORTRAN-callable routines available in the JADE BOS package. More details of each routine will be found in the BOS Report.

In the table * = input variable to the subroutine
+ = output variable from the subroutine

format of a BDW is given in Figure

block-length	reserved
2 bytes	2 bytes

Figure 39. Format of a Block Descriptor Word (BDW)

where:

block-length
is a binary count of the total number of bytes of information in the block. This includes four bytes for the BDW plus the sum of the segment lengths specified in each SDW in the block. (The permissible range is from 8 to 32,760 bytes.)

reserved
is two bytes of zeros reserved for system use.

The format of an SDW is given in Figure 0.

segment-length	code	reserved
2 bytes	1 byte	1 byte

Figure 40. Format of a Segment Descriptor Word (SDW)

where:

segment-length
is a binary count of the number of bytes in the SDW (four bytes) plus the number of bytes in the data portion of the segment following the SDW. (The permissible range is from 4 to 32,756 bytes.)

code
indicates the position of the segment with respect to the other segments (if any) of the record. Bits 0 through 5 are reserved for system use and are set to 0. Bits 6 and 7 contain the codes: 8 9

Code	Meaning
00	This segment is not followed or preceded by another segment of the record.
01	This segment is the first of a multisegment record.
10	This segment is the last of a multisegment record.
11	This segment is neither the first nor last of a multi-segment record.

reserved
is a byte of zeros reserved for system use.

Blocked Records: For blocked records, if the logical record length is less than or equal to the length of the block (allowing four bytes for the BDW and four bytes for the SDW), the block will contain the entire logical record. The next record, preceded by its SDW, begins in the same block (see Figure 40.3).

If the logical record is greater than the length of the block, the record is divided into record segments. The number of segments is determined from the READ/WRITE statement and the BLKSIZE specification. The next record, preceded by its SDW, begins in the same block. If the length of the second record exceeds the remainder of the block it too is segmented (see Figure 40.4).

Example: Assume BLKSIZE=44

```
REAL*8 A,B,C,D
  .
  .
  .
WRITE(9) A,B
  .
  .
  .
WRITE(9) C,D
```

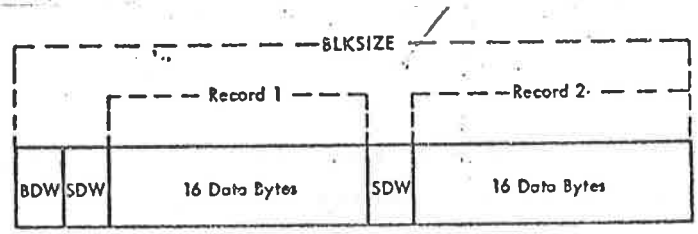


Figure 40.3. Blocked Records Written Without FORMAT Control

Example: Assume BLKSIZE=32

```
REAL*8 A,B,C,D,E,F,G,H
  .
WRITE(9) A,B,C,D
WRITE(9) E,F,G,H
```

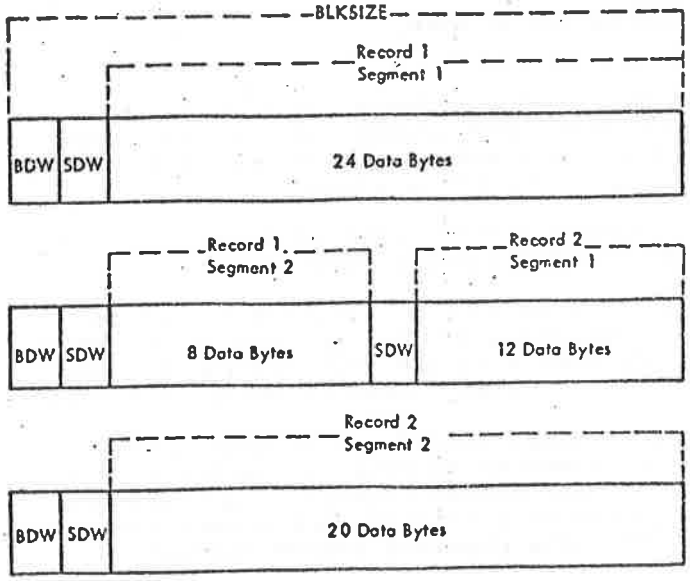


Figure 40.4. Blocked Segmented Records Written Without FORMAT Control

@(J-P)IBM-DATA

INPUT-FILE: M-T-1

OUTPUT-FILE: "file-name"

If the output file does not yet exist, one should create a continuous file before program starts. Ask NORD experts how to do this. If the output file is L-P an octal dump is written to the line-printer (BREAK, if you think it is enough).

RECORD FORMAT ON TAPE: F or VBS

The record length is read from the data on tape.

RECORD FORMAT ON DISK: must be the same as above:

END OF FILE

The data can now be read by the JADE-LIB routine YRVBS.

2. From NORD to IBM

This example shows how to transfer constants from the NORD to the IBM. For more details see the other examples.

a. At the NORD:

Mount an unlabelled tape, press ONLINE.

@(J-P)IBM-DATA

INPUT-FILE: file-name

OUTPUT-FILE: M-T-1

RECORD FORMAT ON TAPE: F

RECORD FORMAT ON DISK: F

RECORD LENGTH ON DISK IN BYTES: 2048* (< 6400)

END OF FILE. COPY ANOTHER FILE: YES or RETURN

b. At the IBM

```
DIMENSION IA(512 )
:
:
READ(1) IA
:
//GO,FT01F001 DD UNIT=TAPE, DSN=something, VOL=SER=volume#,
// DCB=(DEN=3, RECFM=F, BLKSIZE=2048*), LABEL=(,NL)
```

The numbers marked with a * must be the same.

The routine to write VBS records at the NORD is not yet written, but anticipated.

A. IBM-CHAR Program description

1. From IBM to NORD

   The example shows how to transfer program coding written under NEWLIB control from the IBM to the NORD.

a. At the IBM:

   // EXEC NEWLIB, PS='sourcelib'

   ./ COPY $name_1$ [$name_2$] (see NEWLIB manual)

   // COPY DD UNIT=TAPE, DSN=what you like, VOL=SER=volume number,

   // DCB=(DEN=3, RECFM=F, BLKSIZE=80) [, LABEL=(,NL)]

   Instead of ./ COPY one may use ./ EXPAND if one has MACROS. The LABEL parameter is only necessary if the tape is not IBM standard-labelled. The BLKSIZE parameter must be ≤ 136.

b. At the NORD:

   Load the tape on the tape unit, press ONLINE.

   @ (J-P)IBM-CHAR

   INPUT FILE: M-T-1

   OUTPUT FILE: "file-name"

   The file names may be typed in SINTRAN convention. The quotes must be omitted if the file already exists. In this case it gets over-written. If one types L-P on output file one gets a copy on the line-printer.

   LIST INPUT: 10 number of lines to be listed on terminal

   The program checks now, if the tape is mounted, and skips an IBM-Label, if it is there.

   RECORD LENGTH ON TAPE IN BYTES: 80 10-136 possible.
   SKIP COLUMNS 73-80: YES contains useless NEWLIB information.

   The program starts copying. Each byte is translated from EBCDIC-code to ASCII-code. In addition, trailing blanks are removed from the record.

   END OF FILE
   NUMBER OF RECORDS WRITTEN

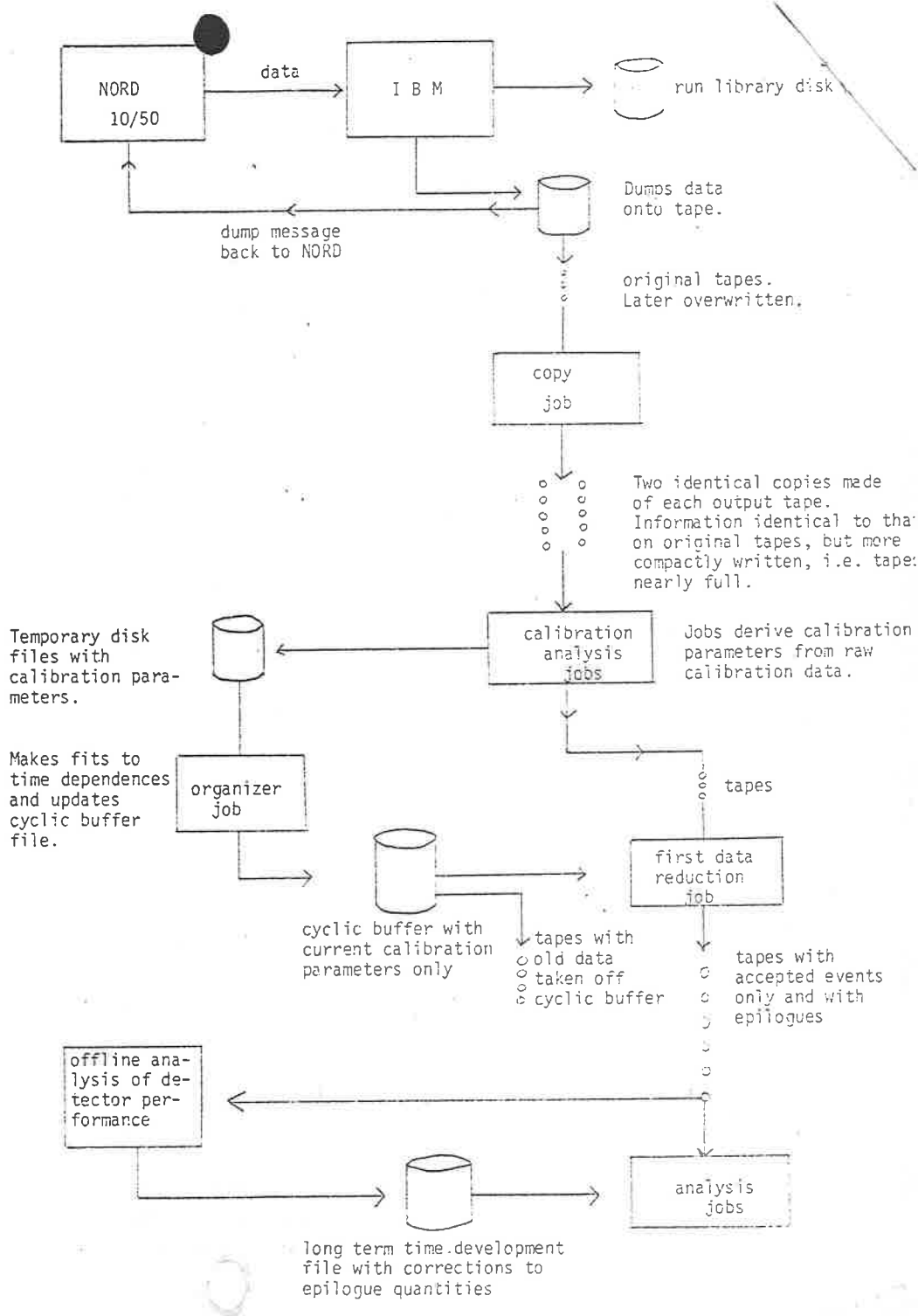
   The file can now be accessed from the NORD editor. If any error messages appear in the program, try to see a NORD expert.

major changes in the apparatus, changes in the running mode of PETRA etc. Many analysis jobs will run on events within one such block of time and they will need less disk space and use less input/output time if the large time development file is split up.

In summary this note suggests the establishment of two disk files for handling calibration data : (1) a cyclic buffer containing fits to the current calibration data needed to write the tape epilogues described above. (2) a large file describing the long term time development of each calibration parameter. The parameters on this file will be for application to corrected data written into epilogues on the first and later generations of reduced data tapes. These corrected data will already contain the calibration parameters as best they were known when the first data reduction job ran. The time development disk file will then only have to parameterize residual corrections. If these are small, smoothly varying in time and applicable only to a small fraction of the 40 000 calibration parameters, this disk file can be manageably sized. The following chart shows how the suggested scheme would fit into the general flow of information from the NORD to the offline analysis jobs. This chart supplements and to some extent modifies the outline on p.2 of the note by W. Bartel, First draft for an IBM data organisation system, dated 26.4.1978.

L. O'Neill will begin to work on testing the feasibility of managing the disk files described above or some generally similar system. Members of the JADE-Collaboration are strongly encouraged to consider how hardware and programming problems would interact with such a system, and to make the appropriate suggestions.

One point bears reemphasis : The above discussion concerns only the administration of calibration data assumed to be available. No one in either the online or offline software group is working on programs to derive the calibration parameters from the raw test run data. The groups responsible for the various hardware systems should consider themselves responsible for providing such programs for whatever calibration parameters their system will require. This applies expressly to those groups, if any, which hope to run such programs on the NORD 10/50, as well as to those which prefer to run at the IBM. Clearly if a tape epilogue system such as the one outlined above is to be adopted, these programs must be ready for the start of the data run.



term "data reduction" refers to the stage at which a given set of events passes through the first offline analysis and is split from the most obvious background events. The term "advanced data analysis" refers to a time when, for example, particle identification is being attempted via dE/dx measurements or when tracks are being fitted for the ultimate in momentum resolution.

Not noted in the table are the numbers of parameters needed to specify each set of data at any one given time. These are often not yet known very precisely. However it is clear that the total number of parameters will be dominated by the ∿ 30 000 analog memory gains and parameters describing the ADCs for the inner detector. It is estimated that a total of ∿ 40 000 parameters will be required to specify the state of the apparatus at any one time.

Most of these data will be available in raw form from special calibration runs during which the flash lamps for the leadglass counters will be pulsed, and/or the test pulsers for the drift chamber electronics operated, etc. For technical reasons concerning the operation of the flash lamps for the leadglass counters, it does not appear feasible to have special calibration events occur continuously and be transmitted one by one, interspersed among the normal events, to the IBM. However the calibration run data will be sent as usual over the hardware link to the IBM and written to the normal raw data tapes, once a day or so.

The calibration data so transmitted may be in the form of individual events, in which case the actual calibration parameters will have to be derived at the IBM, or it may consist of the parameters themselves, already derived at the NORD. In either case the group responsible for each hardware system requiring calibration data is responsible for providing programs to extract the necessary parameters from the raw calibration data. No one in the online or offline software group is developing such programs. The rest of this note concerns only the administration of calibration data assumed to be available in reduced form.

The proposal made here envisages the creation and continuous maintenance of two disk files : one, a cyclic buffer, would contain smooth fits to the time dependence of the various calibration parameters as best they were known, but it would cover only a short interval around the time at which

Type of Data	Time for significant change to occur	First guess available	Known as needed for data reduction	Known with final accuracy
Survey	∿ months	before run	before run	advanced data analysis
Drift velocities	∿ 1/2 day	before run	?	?
Leadglass P.M. gains, ADC params. i.e. pedestals, linearity	∿ 1/2 day	before run	after each calibration run	?
Drift chambers (inner + muon) Analog Memory Gains ADC Pedestals	∿ days	before run	after each calibration run	advanced data analysis
Drift chambers TDC Pedestals	∿ days	before run	after each calibration run	after each calibration run
ADCs and TDCs for TOF Counters	∿ days	before run	after each calibration run	advanced data analysis
Overall JADE Status (Parts not working or not there)	?	after each calibration run	after each calibration run	?
PETRA conditions	∿ minutes	continuously	continuously	continuously

Z = { 0 if z-measurement is doubtful or wrong
1 if z-measurement is OK

T1 = No. of track, the hit is correlated with

$L/_R$ = { 0 if the left solution is used for T1
1 if the right solution is used for T1

$T2$, $L/_R$ } in case of an ambiguous hit, V-particles or crossing tracks

3. Track banks containing track results

- label of program producing the results
- $1/\rho$, ϕ, θ, origin of the track
- χ^2, fit parameters
- ambiguity marker (in case that one cannot distinguish between the left and the right solution two trackbanks are filled with the two solutions and the ambiguity marker is set).
- others

H. GENERAL CONVENTIONS

- There will be only one simple trackfinding routine.
- More than one pattern recognition package for complicated situations are needed.
- There will be only one simple backtracing routine.
- The backtracing routines should be interchangeable.
- Each package doing pattern recognition should work only on groups of cells.
- Decisions on goodness of fits and success of pattern recognition should be done only by the 'SUPERVISOR'.

D. INPUT DATA

A. As a first step the input data are converted to the following format. (It might be that the conversion to this format can be done already during the data taking step on the NORD).

1. HPCELL (ICELL); ICELL = 1,98 : pointer to first hit of cell 'ICELL'.
   - HPCELL (97) = pointer to first location after the last hit information.
   - HPCELL (98) = unused.
   - If a cell has no hits the corresponding pointer points to the location of the next hit.

   => HCELL (I+1) - HCELL (I) = No. of locations used for cell 'I'.

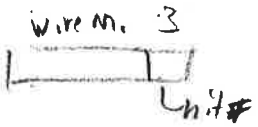
2. HITAR (I) : array with drift chamber data containing always 4 locations per hit :

   IWIR : wire number * 8 + hit #

   A_- : amplitude at '-z'

   A_+ : amplitude at '+z'

   τ : drifttime



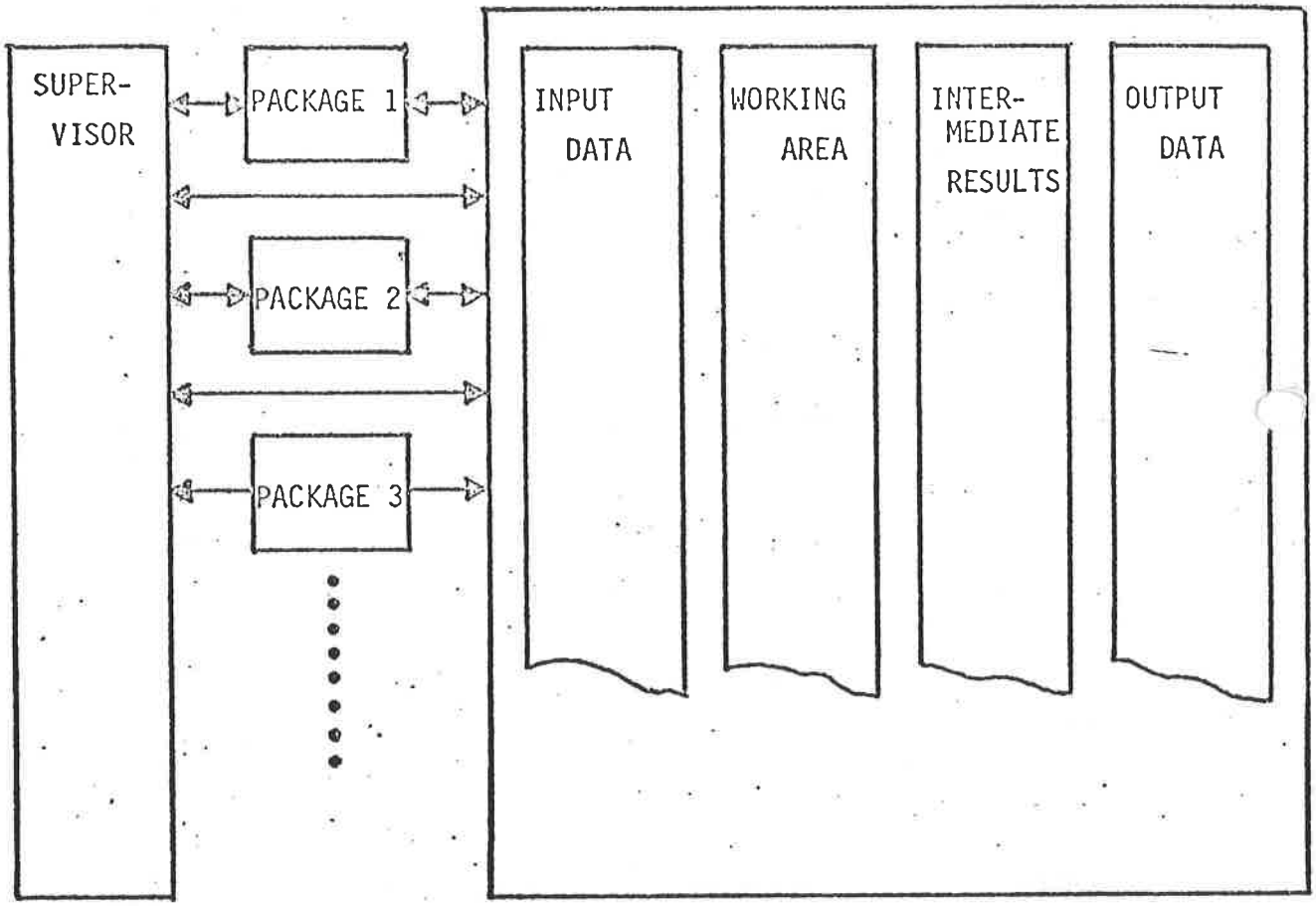
- the number of hits/wire is omitted; the wire number is repeated in case of several hits on one wire.

It is foreseen to overwrite the original data with this format in order to save storage space.

ucal:

.../.

B. Program Structure



SUPERVISOR : program for steering the analysis of one event, replaceable by an operator at an IPS-terminal.

PACKAGE : Package of one or more subroutines for certain tasks in the analysis of an event.
Details are given below.

.../.

Statements of group I are up to 50% faster than the corresponding statements of group II. The first two statements of group I are equal in speed.

B. IBM - NORD10/50 compatability

Whoever thinks his programs could be used at the NORD on-line computer should keep the following remarks in mind.

- PRINT, NAMELIST, RETURN n statements do not exist in NORD-FORTRAN
- hexadecimal constants (Z000F) do not exist
- in-line functions (EXP, SQRT, etc.) are the same, except for all bit- and byte-handling functions (AND, OR, SHIFT, etc.)
- different type declaration statements and default types of variables:

TYPE	IBM	NORD10	NORD50
I - N	32 bits	16	32
A-H, O-Z	32	32	32
INTEGER*2	16	-	-*)
INTEGER	32	16	32
DOUBLE INTEGER	-	32	32
REAL	32	32	32
REAL*8	64	-	-
DOUBLE PRECISION	64	64	64
LOGICAL*1	8	-	-
LOGICAL	32	16	32 bits

*) implementation promised by NORSK DATA (details later)

1 character = 1 byte corresponds to 8 bits in all machines.

Note that the different word length of type INTEGER causes special troubles: the largest value of I in the NORD10 is 32767. Furthermore

```
DIMENSION A(100), IA(100)
EQUIVALENCE (A(1), IA(1))
```

has different effects on the IBM/NORD50 and NORD10. In the NORD10 the IA-array covers only half of the A-array.

II. IPS display routines.

The following display routines are available which can be used for event scanning.

a) R,ϕ view of the detector with tracks and mirror images.

b) R,ϕ view with fitted parabola through high energy tracks, option to clear mirror points of successfully fitted tracks. Option to clear points belonging to a track so that only 'problematic' points are left on the screen.

c) Y,Z view with tracks.

d) Lead glass array unrolled with pulse heights.

e) μ-chamber display.

III. Pattern Recognition.

The following routines are available :

a) The PLUTO pattern recognition program has been modified to the JADE geometry. It recognizes tracks in jet events.

b) Z-vertex finding routine (Steffen scheme).

c) A routine to store track information in a format similar to that described in JADE-Note No. 9a.

IV. Future work.

None of the routines is in its final state and development work is in progress. In case library routines have been copied by outside users, it has to be made sure that they still correspond to the latest version.

Work in the next few months will proceed on the following lines :

a) - improve tracking through the central detector (include multiple scattering and dE/dx losses)
   - update geometrical parameters
   - include end caps.

b) Trigger studies by Monte-Carlo.

c) Set up interactive track fitting with IPS.

b) Test array :

Pointers (40)	ARRAY (max. 12.000)

All quantities are INTEGER*2.

Pointer (1) - points to the last word in ARRAY
Pointer (2) - number of tracks
Pointer (3) - points to the first word of the first track

. .
. .
. .
. .
. .

Pointer (n) - points to the first word of the n^{th} track.

ARRAY :

1^{st} word - wire number
2^{nd} word - drift time
3^{rd} word - R in 0.1 mm
4^{th} word - ϕ in 0.1 mrad (ϕ may be negative or positive)
5^{th} word - Z in mm
6^{th} word - left amplitude.

Word 1 - 6 is repeated as often as there are wires hit for a specific track.

c) Lead glass array :

Pointer (1)	ARRAY (max. 12.000)

All quantities are INTEGER*2.

Pointer - points to the last word in ARRAY
1^{st} word - block number
2^{nd} word - energy in MeV.

I.2.2 Lead glass arrays

a) the energy deposit for photons and electrons is fitted to an empirical shower profile,

b) all other particles are assumed to be minimum ionizing. The energy deposit is 5.83 MeV/cm.

c) Yamada in Tokyo is working on a more sophisticated program to simulate tracks in the lead glass arrays.

I.2.3 Photon tracking.

For photons conversion in the beam pipe and the inner wall of the tank is taken into account.

Electron pairs are produced with correct energy and angular distributions. Photons which convert in the coil and the outer wall of the tank are assumed to continue in the same direction as electrons, however.

I.2.4 Muon filter.

Muons are tracked through the muon filter without nuclear interaction. A brief outline of the procedure is as follows :

a) The chambers and muon-filter concrete blocks are in an approximately correct final configuration.

b) Bending of the tracks is performed in the magnet yoke (inner steel shielding).

c) The tracking is performed in a step-by-step manner with the length of material traversed (concrete, iron) at each step being calculated. Thus it is easy to insert Coulomb scattering interactions and decays.

d) The hits registered are ordered according to some preliminary version of the readout scheme.

Since the TCS internal buffering facility has been moved there is no need to call this routine at the end of a plot.

HDCOPY

The computer now waits for 7 seconds whilst the contents of the screen are copied to the Gould plotter.

TKVEC

If you are writing a program which is overlayed the routine TKVEC must go in the root segment.

ADVANCED GRAPHING II (AG2)

This package is described in the PLOT-10 manual. AG2 uses TCS and provides a higher level of graph drawing facilities. It is expected that AG2 will not be used significantly. Two changes were made to AG2 during implementation due to word length problems.

PLACE

There are two ways of specifying the part of the screen where the graph should go. One is by a Holerith literal e.g. 3HSTD and the other is by a number. In our implementation only the numeric form is available.

OUBGC

Most of this routine has been dummied. It concerns buisness calendars and so will probably not affect us.

CERN HISTOGRAM PACKAGES - ZHIST & ZH2DH

The ZHIST and ZH2DH together with ZGLIB were written by the CERN DD Division. ZHIST provides facilities for one-dimensional histograms - booking, filling, displaying and storing. ZH2DH provides similar functions for two-dimensional histograms. The routines are described in CERN documents. The changes described in "UPDATES TO ZHIST & ZH2DH" have been made. Titles are now specified as character strings instead of arrays.