Most of the tests shown on page 1 should be self-evident but some explanation may be helpful here. Test number 4, for example, checks whether the mass of a given particle agrees with the mass given in the particle data booklet for the given particle type code. The agreement has only to be better than 10 MeV/c^2. Test 5 checks that the Energy, Momentum and Mass are consistent to 5%. Tests 6 and 7 ensure that the production vertex for a given particle is reasonable.

Tests 1 and 13 to 17 are performed on the general event parameters. All the others are performed on every particle that is to be tracked, that is, on every particle that is placed in the VECT/0 bank with the exception of any partons that may be stored at the end.[1] The particles that are placed in the PALL bank are not checked.

Even if only a single particle fails any of the tests, the event is rejected.[2] This is necessary for safety reasons. For every error detected, a 1 line error message will be printed out which indicates which test was being performed and for which event and particle[3] sequence number. The event sequence number might not be the same as the event number created by the 4-vector generator incidentally. After the basic error message, additional information will be printed where appropriate. This will usually be a complete printout of the 10-vector of the bad track.

3. Disabling Tests in Special Circumstances.

When some non-standard data are to be tracked, for example, heavy stable particles, then it will be necessary to disable 1 or more of the tests. This is very easy to do. COMMON / CVFLAG / contains an array with 20 LOGICAL*4 flags which are block data set to .TRUE. . The n'th flag is associated with the n'th test. Thus to disable test 5, do the following in the main program that calls MCJADE:

```
      LOGICAL*4 VTEST
      COMMON / CVFLAG / VTEST(20)
            :
C       Other declarations etc
            :
      VTEST(5) = .FALSE.
            :
      CALL MCJADE(0,1)
            :
```

Please note that there are no tests with numbers 10,18,19 and 20 so manipulating these flags will have no effect. In order to avoid accidents, a warning message is printed before tracking commences if any of the flags are found to be set to .FALSE..
It should be stressed here that the tests performed by MCVALI are very fast and should not normally be disabled.

The TEX source of this note is in 'JADEPR.TEXT(JADECN72)'.

[1] See JADE Computer Note 69, last page.

[2] Except when the only error is a photon of energy $E \leq 200\ MeV$ failing test 5. In this case the event is kept.

[3] For tests 1 and 13 to 17, the particle sequence number is printed as zero ('0')

THE JADE SUPERVISOR 1/02/84
@
JADE COMPUTER NOTE 73

C. BOWDERY
J. OLSSON

****
* *
* The JADE SUPERVISOR Program : An Introduction for Newcomers *
* *
****

A general description of the JADE SUPERVISOR program chain and its detailed working has often been requested, particularly by people who did not partake in the JADE experiment from the very beginning. The following note is an attempt to summarize the general scheme and the most important details.

What is the SUPERVISOR?

The SUPERVISOR is a subroutine (FORTRAN name: SUPERV) that handles most aspects of the standard JADE data analysis for the physicist user. It is the analysis steering routine inside REDUC1 and the graphics program JADEZ and is able to handle all types of JADE events, be they Monte Carlo, partially analysed real data, "fully" analysed TP events or whatever (but not 4-vector events in "CPROD" format !). The SUPERVISOR can also be used to simplify a user's own physics analysis routine by handling the I/O, initialisation and calibration.
@
What is the difference between the SUPERVISOR and the TP Program?

The SUPERVISOR is a more flexible routine than the TP program and has the possibilities to add additional analyses, make cuts and histogram quantities as required. The TP Program however includes analyses that are not done by the SUPERVISOR, can be steered by a small set of control cards and, of course, creates the TP Summary Banks.

How can it be used?

If you simply want to perform or complete the standard JADE analysis for every event on a dataset then life is very easy. To activate the SUPERVISOR you can use the program JDMAIN0, included either in source form by a %MACRO statement or in compiled form by an INCLUDE linkage editor command. Alternatively, you can write your own main program as follows:

```
CALL SUPERV
STOP
END
```

Until voice recognition computers come into common use, we really can't make it any simpler! Additionally you must specify the names of the input and output event datasets (which must be different) and, in the case of real data, the name or names of the JADE calibration files you require . The execution step JCL is shown on page 3.
@
```
;
//*
//*        Default streams: input = 2  , output = 3
//*
//G.FT02F001 DD DISP=SHR,DSN=input.file
//G.FT03F001 DD DISP=NEW,DSN=output.file,SPACE=(TRK,(20,5),RLSE),etc...
//*
//*        Full Calibration Files including Spinning Block Data
//*        Default stream numbers: 21 and 22
//*
```



```
//G.FT21F001 DD DISP=SHR,UNIT=FAST,VOL=SER=STOR05,DSN=F11LHO.BUPDAT0
//G.FT22F001 DD DISP=SHR,UNIT=FAST,VOL=SER=STOR05,DSN=F11LHO.BUPDAT1
//*
;
```

If the events have already had the Lead Glass analysis performed, the short calibration file F11LHO.AUPDAT1 could be used instead of BUPDAT0/BUPDAT1, attached to stream 22. Specifying UNIT= and VOL= parameters may speed up the execution of the job incidentally. They are not essential though.

What can I do if I want something more sophisticated?

This is no problem. First of all though, a few words about how the SUPERVISOR works will help in answering the question.

The routine consists of a series of calls to analysis packages interspersed with calls to a routine called USER. The default version of USER simply passes control back to the SUPERVISOR after each call. By including your own version of USER into the SUPERVISOR job, the program can be tailored to your own needs. The easiest way to start this tailoring process is to copy the basic USER routine to your own library and add code to it where necessary. Alternatively ask around to see what is already available and use or develop an existing, more advanced version of USER. Be careful though that you do not pick up an old version which may not work any longer or may have incorrect comments.
@
At what stages of the analysis are the calls to USER?

The SUPERVISOR execution goes through a series of LEVELS. At each level a particular analysis is performed provided that it has not already been done. That is, if the appropriate result bank(s) already exist then nothing is done. At the end of execution of the code at a given LEVEL, there is a call to USER. The LEVELS are as follows:

LEVEL	Action Performed/Checked BEFORE Call To USER
0	SUPERVISOR initialised at program start
1	New event read in (USER only called if this is a new RUN; otherwise 1st USER call at Level 2)
2	Nothing further has been done
3	Lead Glass energies have been computed
4	Jet Chamber calibrated and Fast z vertex found
5	Jet Chamber Pattern Recognition done + MC Traceback
6	Lead Glass Shower Analysis done
7	Tracks and LG Clusters joined.
8	Nothing further done - empty level at present
9	Muon Chamber analysis done
10	Nothing further done - empty level at present
100	Last event has been written out (On completion of this level, program stops.)

The SUPERVISOR keeps track of the current execution level with a counter called INDEX which is passed to USER as the only argument. Normally, USER performs some operation at a given LEVEL and then increments INDEX
@
before returning control to the SUPERVISOR in order to continue the analysis. Two other levels also exist in the SUPERVISOR but a call to USER does NOT occur at the completion of these. They are:

LEVEL 11	The current event is written out and execution continues at LEVEL 1 or LEVEL 2.
LEVEL 12	Immediate termination of the analysis, statistics are printed out and execution continues at LEVEL 100

Thus LEVEL 11 can be reached simply by incrementing INDEX at the end of LEVEL 10 in USER or by setting INDEX = 11 at any LEVEL. This has the effect of terminating the analysis chain for this event, writing out the

event and starting the next. LEVEL 12 can only be reached by assigning INDEX = 12 in the USER. This is the only valid way of terminating the SUPERVISOR program. Do not set INDEX = 100 as this will result in an error message and loss of SUPERVISOR statistics.
----> Setting INDEX = 1 at any LEVEL in USER will result in the current event being dropped and the analysis started on the next event.

What sort of operations can the USER be programmed to do?

One way the USER can influence the SUPERVISOR is by changing some or all the default values used during program execution, for example, the input, output and calibration logical unit numbers\ or the destination of the program printout. These operations need only be done usually once per program run. Thus the necessary code could be placed in USER at the point set aside for servicing the initialisation call (LEVEL 0 i.e. after the SUPERVISOR has initialised itself). This level should be used for booking histograms, reading in special options, printing banner headlines etc., etc.
@ Secondly, by manipulating the INDEX variable that the SUPERVISOR uses, certain analysis calls can be skipped or even done out of sequence although the latter is not always feasible or desirable because of the interdependence of some of the analysis steps. Additionally, decisions about proceeding with the current event or rejecting it or stopping the analysis altogether can be made at any level and carried out by setting INDEX to one of the special values provided for these purposes. A word of warning is necessary here. Care must be taken when changing INDEX to ensure that infinite loops do not occur or similar disasters.

Thirdly, histograms can be filled at various levels after the analysis packages have been called. A call to HISTDO at LEVEL 100 in USER is an ideal way of printing the histograms at the end.

Fourthly, by deleting results banks, a re-analysis of a detector part can be performed. For example, by deleting the PATR and JHTL banks at the call to USER at the end of LEVEL 4 (or earlier), the PATREC package will create a new one with the latest constants at LEVEL 5. By deleting the muon results bank MUR1/0 and/or MUR2/0 before LEVEL 9, a new muon analysis will take place.

Where can I find the SUPERVISOR and the analysis packages it calls?

The SUPERVISOR, default USER, JDMAIN0 and many important JADE routines (in source/compiled form) reside on F11LHO.JADEGS/JADEGL. Other routines can be found as follows:

Lead Glass analysis	.....	JADELG.SOURCE/LOAD
I.D. Pattern recognition	.....	F11GOD.PATRECSR/PATRECLD
Muon analysis	.....	F22ALL.JADEMUS/JADEMUL
BOS routines	.....	F1EBLO.BOSLIB.S/BOSLIB.L

@
Appendix A SUPERVISOR Outline for the Inquisitive

The following is an outline of SUPERVISOR actions, in the order they occur between the various calls to USER. Note that all arguments in subroutine calls have been omitted for clarity. Further details can be obtained from the source listings.

The JADE standard BLOCKDATA, with most of the geometrical constants, is linked via the statement:

```
EXTERNAL JADEBD
```

BOS initialisation (currently with COMMON / BCS / IDATA(40000)) :

```
CALL BINT
CALL BWRO
```

Initialization of various analysis routines. Note that VTXINI is called, although the vertex finding programs are not yet called in SUPERV.


```
CALL LGINIT       (Lead glass analysis)
CALL INITZV       (Z-vertex finding)
CALL MUINI        (Muon analysis)
CALL VTXINI       (Vertex finding routines)

    ---->>  CALL USER(0)
```

Start event loop; first the remaining CPU-time is checked with the function

```
IUHR(ISECLF)
```

@
ISECLF is found in COMMON /CSECLF/ ISECLF and is BLOCKDATA set to 2. It could be changed at level 0.

Read event:

```
CALL EVREAD
```

Note that EVREAD automatically handles the special constants records in the beginning of every MC data file (see JCN 66). After reading, a copy of the bank HEAD is stored in COMMON /CHEADR/ HEAD(108). Note here that the first word of the bank is found in HEAD(9). This is an Historische Eigentuemlichkeit of JADE. Thus the Run and Event numbers are found in HEAD(18) and HEAD(19) respectively. Note also the standard JADE convention, with IMPLICIT INTEGER*2 (H).

IF the read event is the FIRST event of a NEW Run:

```
CALL KALIBR       (Calibration constants)
CALL INPATC       (Init. Pattern Recognition)
and for the very first event:
CALL INPATR       (Init. Pattern Recognition)
```

Thus initialisation of PATREC takes place after the calibration files have been read. Note that INPATR is called only once, while INPATC is called at the beginning of each new run, to take account of e.g. changes in Lorentz angle, etc..

```
    ---->>  CALL USER(1)  (new run only)
```

For ALL events, before any analysis has been done/checked:

```
    ---->>  CALL USER(2)
```

@
Lead glass calibration, if the bank ALGN does not yet exist:

```
CALL LGCALB
    ---->>  CALL USER(3)
```

Jet chamber data calibration, if the calibrated JETC bank does not yet exist; immediately followed by the fast Z-vertex finding (bank ZVTX):

```
CALL JETCAL
CALL ZVERTF
    ---->>  CALL USER(4)
```

Pattern recognition, if no bank PATR exists or if only bank PATR 12 is present (MC events). PATREC result banks PATR and JHTL are created. The Monte Carlo backtrace facility is also prepared here, with the creation of the bank TR4V (JCN 69):

```
CALL PATREC
IF( M.C. ) CALL MCTR4V
    ---->>  CALL USER(5)
```

Lead glass cluster finding. The LGCL bank is created:

```
CALL LGANAL
    ---->>  CALL USER(6)
```

Charged track - LG cluster connections and photon energy corrections; The bank LGCL is partly overwritten with new results, see JCN 14C.

```
                        CALL LGCDIR
                            ---->>  CALL USER(7)
@
At level 8 nothing happens at present, free for later use:

                            ---->>  CALL USER(8)

Muon analysis. MUR1 and MUR2 banks created.

                        CALL MUANA
                            ---->>  CALL USER(9)

At level 10 nothing happens at present, free for later use:

                            ---->>  CALL USER(10)

At the end of the event loop, the event will be written out, unless  the
INDEX variable has been set to the value 1, with  subsequent  new  event
reading:
                        CALL EVWRIT

At the end of the job (EOF encountered, TIME LIMIT or USER forced),  the
last part of an event is written  out,  statistics  over  accessed  USER
levels as well as event statistics are printed  and  in  addition,  muon
statistics (if available) are also printed:

                        CALL BMLT
                        CALL BWRITE
                        CALL MUFINI
                            ---->>  CALL USER(100)

Finally a RETURN to the MAIN program is done.


    Type:  SUB  'JADEPR.TEXT(JADECN73)'   to get a copy of this note.
/*
```