The EOSGX manual for TAS 2

Mark Gilkes

State University of New York, Stony Brook

and

A. Dean Chacon

Texas A & M University

August 6, 1997

Abstract

Eosgx is a program that interfaces the EOS TPC analysis shell (TAS), and GEANT. This manual describes the commands that EOSGX will accept, and how some of the non-standard features were implemented. It is assumed that the reader has access to the GEANT 3.15 manual [1], the GEANT 3.21 manual [2], and to the PAW manual [3]. The last revision date for this manual is 20-Dec-1995.

This manual is a working document, and it is the opinion of the authors that it will remain so for the useful life of the program it describes.

Contents

1 Introduction

> ... the Hitch-hiker's Guide has already supplanted the great *Encyclopaedia Galactica* as the standard repository of all knowledge and wisdom, because although it has many omissions, contains much that is apocryphal, or at least wildly inaccurate, it scores over the older more pedestrian work in two important ways. First, it is slightly cheaper; and second, it has the words "DON'T PANIC" inscribed in large, friendly letters on its cover.
>
> — Douglas Adams in *The Hitch-hiker's Guide to the Galaxy*

EOSGX is a program which combines the GEANT package of simulation codes with the EOS TPC analysis shell (TAS). The 3.15 version of EOSGX links to version 3.1590 of GEANT and the V93c version of the CERN library. The 3.21 version of EOSGX links to version 3.21 of GEANT and the V94a version. The 3.15 code is maintained in CMS library EOSU:[EOSG.CMS].[1] The 3.21 code is maintained on the EOS cluster in CMS library EOSU:[EOSG_321.CMS].[2] There are also TAS2 versions of EOSGX maintained using CVS on the UNIX machine SSEOS. EOSGX is appropriate for use in any EOS simulation study involving GEANT, as it contains the full GEANT code, apart from a few interactive GEANT menus (see below). The program has also been specifically tailored for studies of the track reconstruction efficiency in the TPC analysis chain (see Section 2).

In order to implement the GEANT/TAS combination, a few key modifications were necessary. The main program of EOSGX initializes GEANT, PAW and TAS so that the ZEBRA structures for each occupy different stores, so that the codes do not share memory except for the special case of the PAWC common, which has historically been referenced by GEANT. The interactive portion of GEANT contains histogramming and plotting options which become redundant in a GEANT + PAW interface — so that in most such cases the GEANT references have been removed. This means, unavoidably, that there are modified GEANT subroutines in the EOSGX library which have to be updated along with any GEANT version updates.

There are also minor modifications to some the action routines of the GEANT drawing menu to correspond with features of the EOSGX event display, and to the GIDROP routine

[1] The location of the library is as of 25-Oct-1994.
[2] The location of the library is as of 25-Oct-1994.

used for clearing the geometry banks. These modifications increase the flexibility of EOSGX at the KUIP command level. All PAW, TAS and unique GEANT commands are available, in addition to a special menu, EOS_GEANT, particular to EOSGX.

The connection between GEANT and TAS is established in a modified version of subroutine TEP_AM, in which special analysis modules have been defined whose only action is to initiate a GEANT event loop at particular points within the TAS GO loop. These analysis modules are switched on and off internally by EOSGX when appropriate, and can therefore be ignored by the user in most cases. Other analysis modules are more conventional in purpose: the GEANT_TRACKS module outputs tables with kinematics from the simulation, and the GEANT_DISPLAY module draws hits on line within the TAS event loop.

EOSGX currently has the ability to (1) analyze experimental data, and (optionally) then pass the results of the data analysis (positions, momenta and particle ID) to GEANT so that the results of the simulation may be compared to experiment, and (2) read a FREESCO (or other event generator) data file, pass the particle momenta and ID to GEANT to simulate the detector response, and then (optionally) pass the data to the data analysis routines so that the results of the analysis may be compared to the input to the simulation.

1.1 Notation

Program and subroutine names will be written in SMALL CAPITALS. TAS table names will be written in CAPITALS. Command names and their arguments will be written in CAPITALS. Elements of the EOS detector system (e.g., the TPC and MUSIC) will be written in CAPITALS. In examples of command lines, the prompt for the program (TAS>) will not be shown.

1.2 Acknowledgements

The authors wish to acknowledge that sections of the codes were written by M. Justice, W. Müller, D. Olson, G. Rai, and H. Wieman. We also wish to acknowledge the comments from the users of earlier versions of the manual.

2 Program layout

2.1 Geometry handling

In EOSGX the physical specification of individual detectors is coded into the geometry routines, as the detectors already exist and are not expected to undergo further development. However, the positions of the detectors in the Master Reference System (MARS) are not hardwired in this way.

2.1.1 Geometry input source

Volume sizes, positions and orientations are stored in the GEANT ZEBRA data structure JVOLUM (see the GEANT [2] manual for a description of JVOLUM). In EOSGX the position and orientation of the major detectors may be loaded into JVOLUM from two possible sources: the FFREAD file, or the TAS ROTM table.

The GEOMETRY command determines which source of positioning information is to be used. The command

GEOMETRY FF

specifies the FFREAD file, while GEOMETRY TT (the default) selects a special TAS table, the ROTM table.

The two choices have a different functionality. The FFREAD file is read by the program during initialization, so that the position data cannot be modified afterwards without re-running the program. This is not a problem if the detector positions are well-known, and are not under study. However, this method of positioning the detectors has generally been superseded by the use of the ROTM table.

The handling of the ROTM table information is dynamic. If the positioning source is set to TT, the GEANT geometry reflects the contents of the ROTM table at all times. If the ROTM table is modified, the changes are automatically loaded into the JVOLUM data structure. If the data structure is modified by means of the MOVE command, the ROTM table is automatically updated to correspond with the new positions.

2.2 Run modes within EOSGX

EOSGX contains the full GEANT code used for event simulations. The relationship between the GEANT event loop and the data (TAS) event loop is determined by the choice of a run mode. There are four run modes: NORM, DSIM, FSIM and DATA. A particular run mode is established with the command RUN_MODE (see Section 3.4).

1. NORM – In the NORMal mode, the GEANT event loop is separate from the TAS event loop. The two halves of the EOSGX program exist in parallel, but there is no event processing link between them. The only connection of any kind is through the display software (see event displays). In this mode, the GEANT code runs as a conventional simulation.

2. DSIM – In this mode the GEANT event loop is called from within the TAS event loop before data analysis has started, in order to produce simulated TPC and TOF hits for the analysis. The TPC hits are written into the TAS HITS table and are input to the tracker for studies of the efficiency of the tracker. The TOF hits will be used to study track matching.

3. FSIM – In this mode, the GEANT event loop is called in order to provide kinematics input to the pad response simulator. The output is either written to a *.BDF file or to the DTS_RAW table directly. There need not be coupling between the TAS and GEANT event loops. However, this mode is unique in that it requires a special initialization of the TPC geometry appropriate for the pad response simulator.

4. DATA – In this mode, the kinematics input to the GEANT event loop is the TRACKS table of the TPC and MUSIC detectors. A TAS event is recast as a GEANT event for the purpose of diagnostic track matching between the TPC, MUSIC and TOF detectors. In this mode the GEANT event loop is called from within the TAS event loop after data analysis has filled the tables.

Some of the run modes require that certain TAS modules be enabled and others disabled for the run modes to work properly. The enabling and disabling of these modules is done automatically, and a list of these modules and others is given in Section 4.

2.3 Input modes within EOSGX

The EOSGX input modes determine the method of input for the particle kinematics and species information. This forms the simulated physics event input for the GEANT part of the simulation. There are six input modes, ONE, MANY, INTG, FILE, DATA, and PRTS. A given input mode is selected with the command INPUT_MODE, for example

INPUT_MODE FILE

sets the input mode to FILE.

Each input mode is accessible to any of the run modes of the program (see Section 2.2 for a description of the run modes), except the DATA input mode. DATA input mode is the only possible choice for the DATA run mode. The DATA input mode can also be used in NORM run mode, but not in the simulated data modes DSIM or FSIM.

2.3.1 Convention for the particle ID's

EOSGX adopts a convention by which all particle types known to GEANT are to be determined by the program on the basis of the particle Z and A only. The convention used is as follows:

All nucleons and nuclei:	Z = charge	A = mass number
All pions:	Z = charge	A = 0
All other particles:	Z = GEANT PID	A = −1

The particle ID's are assigned internally for all nucleons, nuclei and pions. For all other particles, the particle type must be read in by EOSGX in place of the charge. The charge is then assigned internally. For the list of GEANT particle ID's, see the GEANT manual [1].

2.3.2 Description of the input modes

1. Input mode ONE

   In this input mode, intended for diagnostic purposes, a single particle is entered explicitly into the program. The event will consist of this single particle only, unless secondaries are generated. The particle type, kinematics and vertex information are entered by using the PARTICLE, PKINE and ORIGIN (or GVERT) commands respectively (see also these commands in Sections 3.1 and 3.2). For example, the commands

```
INPUT_MODE ONE
PARTICLE 79 197
PKINE HTPC 1. 0. 0. E
ORIGIN TARG
TRIG 5
```

specify a gold nucleus with kinetic energy per nucleon of 1.0 GeV originating in the geometrical center of the target and traveling downstream along the TPC axis. Five GEANT events are requested — each is identical, in the absence of optional random physics processes. As another example, the commands

```
INPUT_MODE ONE
PARTICLE 10 20
PKINE MUSI 0. 0. 100. P
ORIGIN MUSI 0. 0. -125.
```

specify a neon nucleus with momentum $p = p_z = 100$ GeV/c in MUSIC coordinates originating at a point 125 cm upstream of the MUSIC center and traveling downstream along the MUSIC z-axis.

2. Input mode MANY

This input mode is functionally equivalent to input mode ONE, except that multiple particle definitions are possible (with the same sequence of commands to specify a particle as described for input mode ONE). This is once again a diagnostic feature, since the content of the multi-particle event is fixed for all events, unless smearing of the event vertex or physical processes are in effect. As an example, the commands

```
INPUT_MODE MANY
PARTICLE 1 1
PARTICLE 2 4
PARTICLE 3 6
```

```
PKINE MARS 0. 0. 1. P
PKINE MARS 1. 0. 0. E
ORIGIN FREE
GVERT SVTX 0. 0. 1. 0.5
```

specify an event containing one proton of momentum $p = p_z = 1.0$ GeV/c, one ^{4}He particle with 1.0 GeV kinetic energy per nucleon and one ^{6}Li particle also with 1.0 GeV kinetic energy per nucleon. The three particles originate from a vertex located within the target and smeared, event by event, with a mean corresponding to the target center and a Gaussian distribution in x and y in target coordinates with VSIGMAX = 1.0 cm and VSIGMAY = 0.5 cm. The smearing in the third coordinate z is uniform throughout the target thickness. (See also Section 3.1 for a description of the vertex specification.)

The scheme used to build the multi-particle event assumes that the multiplicity of the event is the number of entries in the PARTICLE or PKINE lists, whichever is greater, that have been input since the last INPUT_MODE MANY command (which has the effect of clearing the lists). The missing information in the shorter of the two lists is filled in from the last entry in that list. Therefore the commands

```
INPUT_MODE MANY
PARTICLE 1 1
PKINE MARS 1. 0. 0. P
PKINE MARS 0. 1. 0. P
PKINE MARS 0. 0. 1. P
ORIGIN FREE
GVERT SVTX 0. 0. 1. 0.5
```

will result in an event containing three protons with 1.0 GeV/c momentum each — one traveling parallel to the x axis, one parallel to the y axis, and one parallel to the z axis in CAVE coordinates.

3. Input mode INTG

This input mode invokes the generators internal to EOSGX. The current internal generators are very simple, and are not intended as alternatives to external event generators based on detailed physical models. The internal generators do, however, have features which make them useful in diagnostic applications. There are currently three internal generators, numbered 1, 2, 3. The default is generator number 1. To choose a different generator, the number should be entered explicitly. For example,

```
INPUT_MODE INTG 3
```

invokes generator 3. The scheme for the internal generators is designed in this way to make the definition of new generators straightforward.

Internal generator number 1 is a simple code that assigns a constant kinetic energy per nucleon to each of N particles. The N particles all have the same particle type and are assumed to emerge isotropically in the reference frame of a fictitious remnant nucleus. The appropriate sequence of commands for this generator would be, for example,

```
INPUT_MODE INTG 1
PARTICLE 2 4
PKINE MARS 1.1 0. 0. E
PKINE RAND 10 20 5. 15.
ORIGIN TARG
```

The kinematics for the remnant are taken from the first PKINE command. The fragments (all ^{4}He's, from the PARTICLE command) will emerge from a remnant system with a kinetic energy per nucleon of 1.1 A GeV traveling downstream along the CAVE (MARS) z axis. The second PKINE command, with the special qualifier "RAND", specifies the limits on event–by–event smearing of the event multiplicity N and kinetic energy per nucleon of the fragments in the reference system of the remnant. In the above example, the multiplicity of fragments is smeared uniformly between 10 and 20, and the kinetic energy per nucleon of each fragment in the remnant system is smeared uniformly between 5 and 15 MeV. The fragments emerge from a vertex at the target center.

Internal generator number 2 is a code that steers N particles, all of the same particle type, into a certain solid angle in the CAVE reference system. The specification for this is taken entirely from the KINE command (or the KINE card in the FFREAD file). See also the description of the KINE command in Section 3.2. The commands

```
INPUT_MODE INTG 2
ORIGIN PLT1
KINE 1 0. 2. 0. 360. 1. 1.2 30 2. 4.
```

specify 30 ^{4}He's with kinetic energy per nucleon between 1.0 GeV and 1.2 GeV emerging from a vertex in the center of the first PLUTO and moving downstream into the solid angle defined by the limits THETAMIN = 0.0, THETAMAX = 2.0, PHIMIN = 0.0, and PHIMAX = 360.0 in the CAVE coordinate system.

Internal generator number 3 was historically intended as a predictive tool for fragment trajectories. Like generator number 1, it uses the PKINE command entries to determine the velocity components of the remnant in the laboratory. Unlike generator number 1, the PARTICLE command does not denote the particle type of the fragments. Instead the PARTICLE entries are used to specify the charge to mass ratio of the remnant, ZREM/AREM, which is used to limit the sum of fragment charges. The actual fragment types themselves are determined from the MASSES command. The command

```
MASSES A1 A2 A3 A4 A5 ...
```

specifies the mass numbers of a series of fragments. The charge of each of the fragments is determined on the basis of natural abundances. Above Z = 6, the most frequently occurring isobar is used. For Z = 6 and below the choice between isobars is randomized and weighted according to abundance (stability). The sum of all fragment charges must obey the following relation:

$$\frac{\mathrm{ZSUM}}{\mathrm{ASUM}} = \frac{\mathrm{ZREM}}{\mathrm{AREM}} \tag{1}$$

The slope parameters in the kinetic energy distribution of fragments are chosen from p-nucleus data. The angular distribution is isotropic in the rest frame of the remnant. However, total energy and momentum are not conserved. The sequence of commands

INPUT_MODE INTG 3
PARTICLE 79 197
PKINE MARS 1. 0. 0. E
MASSES 1 2 4 4 6 20
ORIGIN TARG

specifies an event containing six fragments emerging from a remnant system moving along the CAVE z-axis at 1.0 GeV per nucleon. The charges, and hence the particle types, of the fragments are computed quantities. The ratio of ZSUM/ASUM is assumed to be that of gold, 79/197.

4. Input mode FILE

This input mode is the default input mode for EOSGX. In input mode FILE, EOSGX gets simulated physics events from an unformatted (non-ASCII, machine readable) file, which allows reading of simulated events from an external event generator such as FREESCO, ISABEL, ARC, and etc. The file, historically referred to as an FRS (File Read Source) format file, is given the extension *.FRS. The FRS format is described later in this section. As FRS files are not user-formatted, they are not portable between platforms and must be translated to ASCII for copying from one platform to another.

The designation of a particular FRS file on disk or tape as the current input to EOSGX is handled by the command LUN (Section 3.6) i.e.

LUN unit filename O

opens a file named FILENAME on the FORTRAN logical unit UNIT, and makes UNIT the current input unit. If no file name is supplied, the logical unit UNIT is assumed to already have a file opened on it, and the current input unit is set to UNIT.

The command

LUN unit ! C

closes the file on logical unit UNIT, and deassigns UNIT.

EOSGX maintains a list of those FORTRAN logical units opened for FRS input. While many FRS files may be open at the same time, only one file is designated to be the current source of input. This file is chosen by selecting its associated FORTRAN input logical unit with the command LUN. The sequence of commands

LUN 51 MYFILE_1.FRS
LUN 52 MYFILE_2.FRS

opens two files MYFILE_1.FRS and MYFILE_2.FRS on units 51 and 52 respectively. (The third parameter "O" need not be given, as it is the default.) Because unit 52 is assigned last, it becomes the current source of input and the next GEANT event will come from MYFILE_2.FRS.

The sequence of commands

LUN 51 MYFILE_1.FRS
LUN 52 MYFILE_2.FRS
LUN 54 MYFILE_3.FRS
LUN 52

opens three files on units 51, 52 and 54, then assigns unit 52 to be the current input.

At initialization time, EOSGX will attempt to open default units for files referenced by the VMS logical names (or UNIX environment variables) EOSG_FRS_1, EOSG_FRS_2 and EOSG_FRS_3. To use this default option, these names should be assigned to existing files. The FORTRAN units chosen will normally be 48, 49 and 50 unless these are already in use, in which case the nearest available units will be chosen. The last unit successfully assigned becomes the current source of input.

The read format for input mode FILE is as follows:

First a character string identifying the file. This may have a maximum length of 80 characters.

Read format for the character string is:

READ(LUN_INPUT)	character_string

Then the run header.

EOSGX reads from various logical unit numbers. These units may all be open at the same time, and each of the files associated with these units may have a different number of optional parameters. These differences are accommodated internally. The maximum number of optional parameters is set in the include file EOSG_MGC.INC.

L_PRUN	Number of run parameters for current input
L_PEVT	Number of event parameters
L_PTRK	Number of track parameters
R_PRUN(L_PRUN)	Array of run parameters
R_PEVT(L_PEVT)	Array of event parameters
R_PTRK(L_PTRK)	Array of track parameters

Additional parameters may vary as to their interpretation, depending on the particular event file. The meaning of the optional parameters is to be defined by the application.

Read format for the run header is

READ(LUN_INPUT)	l_prun, l_pevt, l_ptrk, (r_prun(l_zp), l_zp = 1,l_prun)

Then the event header.

K	Event number
NPTRAK	Number of particles in the generated event

The event number is not really necessary, but remains in the code for compatibility with previous event files.

Read format for the event header is

READ (LUN_INPUT)	k, nptrak, (r_pevt(l_zp), l_zp = 1,l_pevt)

Finally the particle data block.

IA(I)	Mass number A of Ith particle (see Section 2.3.1)
IZ(I)	Charge Z of Ith particle (see Section 2.3.1)
PARTGE(1,I)	P_x of Ith particle in physics frame of event
PARTGE(2,I)	P_y of Ith particle in physics frame of event
PARTGE(3,I)	P_z of Ith particle in physics frame of event

Read format for the particle data block is

```
DO I = 1,NPTRAK
READ(LUN_INPUT)   ia(i), iz(i), partge(1,i), partge(2,i), partge(3,i),
                  (r_ptrk(l_zp), l_zp = 1,l_ptrk)
END DO
```

Note that the GEANT particle ID is not a part of the format.

5. Input mode DATA

In this input mode, the event that is input to the GEANT simulation consists of the current contents of the TPC TRACKS table i.e., the output of the TPC analysis code is fed back into the simulation. This input mode is required in DATA run mode. The DATA input mode has been used in the event display to compare the TPC track projections (using command DDTRACKS, for tracks found without taking energy loss into account, and where the track projection is generated without energy loss) with the GEANT track projections for these same tracks (generated with energy loss). This comparison is useful as a check on the effect of energy loss on the trajectory, and is especially instructive when combined with a comparison of the initial GEANT trajectories of the tracks originally input to the simulation.

6. Input mode PRTS

In this input mode, the event consists of the contents of the PARTS table filled by FREESCO. The scheme used to interpret the particle ID information in the PARTS table depends on the usual Z and A convention used in the other input modes (Section 2.3.1), despite the presence of an explicit GEANT particle ID in the PARTS table.

2.4 Output modes within EOSGX

The concept of an output mode in EOSGX is tied to the behavior of the run modes. Unlike the input modes, which are more or less independent of the choice of run mode, the output mode is really an integral part of the run mode functionality. With two important exceptions, the output modes are explicitly tied to the run mode assignment, and are more descriptive than indicative of new program behavior.

In practice, the choice of an output mode is only of importance in the DSIM and FSIM run modes. These are the two modes that produce simulated TAS event output from GEANT; i.e. these modes cause certain data analysis tables such as the HITS table to be filled by the GEANT part of the simulation code. The data tables filled by GEANT then become the input for other data analysis routines, such as the TPC tracker. A standalone simulation (NORM run mode) does not write TAS table output at all, and is automatically assigned to the NORM output mode. In READ run mode, the output is always (and only) to the GEANT_TRACKS table, so that READ run mode ignores an output mode assignment.

However, two choices are available in the DSIM and FSIM run modes, which output to TAS data tables (the HITS or DTS_RAW tables) or to a *.BDF data file, in the case of FSIM mode. These choices are:

1. OUTPUT_MODE WRITE — In WRITE output mode the TAS table or *.BDF file is filled only by the simulation codes.

2. OUTPUT_MODE UPDATE — In UPDATE output mode the TAS table or *.BDF file contains MERGED real data and simulated data. The real data comes from a data file that is currently open, and being read in by TAS.

2.5 Event displays in EOSGX

The interactive tools provided by the GEANT package for drawing a detector setup have been extended in program EOSGX to allow displays of event data from the TAS tables. The display software provides an interface to the GXINT drawing routines. With a few commands, a wide variety of views of the experimental setup can be generated, and the event data displayed.

2.5.1 Displayed data

The displays of event data, since they depend only on the TAS event loop and not on the GEANT event loop, are independent of the choice of GEANT run mode. The display buffers are updated at the end of every TAS GO loop if the analysis module GEANT_DISPLAY is turned on.

Hits may be displayed for any detector. For the EOS TPC and MUSIC, tracks may be displayed. In addition, MUSIC TKS planes may be drawn, as well as the event vertex from the vertex finder, and the beam vector from the PLUTOS. (See Section 3.5 for the list of display commands.)

2.6 SETVIEW

The GEANT drawing routines allow for the generation of arbitrary views of the volumes in the geometrical tree. However, the specification of a particular view may involve many parameters, which must be carefully chosen. Specialized views may be difficult to specify, or may involve a long series of commands.

The SETVIEW software provides an interface to the GEANT drawing package for the EOSGX event display. Simple views of the geometrical setup may be obtained with a minimum of user-supplied parameters. For specialized views, a unified approach has been used in which the parameters of a new view may be specified with reference to the coordinate system of any GEANT volume. This gives the user a means of describing a view orientation in the simplest possible way.

2.6.1 Drawing set views

The SETVIEW command draws a volume or volumes in a particular orientation on the screen. As with the DRAW command in the GEANT menu /DRAWING, the subroutine GDRAW is called to perform the actual drawing. All volumes present in the geometrical tree are available in each of these named orientations, referred to as "set views". Six standard views are defined by default: PERSpective, TOP, FRONt, BACK, SIDE, and ISOMetric (a reverse perspective). An additional view, named AUTO, allows the user to scroll through the standard views of the major detectors, by transferring control to the mouse. It then becomes possible to bring up new views by clicking with the mouse on the centers of main detector volumes as they are displayed. Clicking in the lower left corner of the screen exits this feature.

In views drawn by SETVIEW, the origin and the scale of each view are automatically computed, and the view is drawn with color and visibility attributes for each volume chosen according to internal lists. All view parameters may be modified by the user, with the OPTVIEW command (see section 3.5).

The user may also define additional set views by choosing an orientation of the line of sight and a rotation in the screen plane (a rotation about the line of sight). The angles defining the line of sight are the same as those used by the GEANT DRAW command. The user may choose any names for these new views (other than the seven default names). The current view definitions may be displayed with the command INFO VIEW.

2.6.2 View modes

The optional parameter VCHID, the view mode ID, specifies whether a view will be drawn in global (MARS) coordinates, or in the local coordinates of the volume, or will be executed in BUILD mode. BUILD mode assembles a view for a requested mother volume, one descendant volume at a time, in MARS coordinates, under the control of the graphics cursor. Clicking on the screen with the mouse within the volume boundary draws the next descendant volume, while clicking in the lower left corner exits the feature.

View mode character IDs:

Global = “G”
Local = “L”
Build = “B”

At initialization time, the view mode is set to GLOBAL. The view mode is retained for each subsequent view until a different view mode ID is entered explicitly, in a new SETVIEW command.

2.6.3 The SETVIEW view list

The SETVIEW code maintains a list of volumes which “belong” to the current view. Normally the list is cleared before each new view, but if CLEAR has been turned OFF (see OPTVIEW, section 2.7) the view list becomes the sum of successive set views. This list of volumes can be treated as an entity, by referring to it as a pseudovolume named LIST. The BUILD view mode can also be used to create a view which contains a partial set of volumes from a particular mother volume. This set can then be manipulated by using the volume name LIST (e.g., SETVIEW LIST PERS).

2.7 OPTVIEW

The OPTVIEW command (see Section 3.5) allows for the customization of the set views. The form of the command is:

OPTVIEW option qualifier(s) .

The various options and qualifiers provide for a wide variety of possible choices. Some of these choices, such as those for altering color or visibility attributes, are also accomplished in the GEANT drawing menu /DRAWING by the SATT command. However, the OPTVIEW command operates on the internal lists used by SETVIEW, and is required if changes to view attributes are to be recognized by the SETVIEW code. Current options available under OPTVIEW are: ADD, COLOR, CLEAR, DEL, HITS, LIST, NAME, ORIGIN, PADS, SEEN, TRKS, and UNSEEN. The various options have different qualifiers depending on the particular option selected. These are listed below:

Option	Qualifier(s)	Meaning
ADD	Name of a new view followed by angles defining the line of sight and the name of a system in which the angles are defined	A user-defined view may be added to the list of known views. The view name is a unique character*4 name. The three angles are the polar angle THETA and the azimuthal angle PHI which define the orientation of the line of sight, and the angle PSI which specifies a rotation in the screen plane about the line of sight. An additional parameter, VREF, specifies the reference system in which these angles are defined (the default is 'CAVE').
COLOR	Volume name followed by color index	Change the color of a volume in the set views. The color indices are: 1=BLACK(WHITE), 2=RED, 3=GREEN, 4=BLUE, 5=YELLOW, 6=MAGENTA, 7=CYAN. All other choices give the screen background. These choices affect all subsequent views. A special feature is the ability to render all volumes the same color, using the volume name CAVE. This can be useful when making black-and-white Postscript files, in which certain colors may appear to the printer as white.
COLOR	RESET	Restore default SETVIEW colors (after setting all volumes the same color with the command OPTVIEW COLOR CAVE [color index]).
CLEAR	ON (OFF)	Turn ON or OFF the automatic clear screen before a set view. The default is ON.
DEL	Name of a previously defined view	Delete a view previously defined by the user. (The standard views PERS, TOP, FRONt, BACK, SIDE and ISOM cannot be deleted.)

Option	Qualifier(s)	Meaning
HITS	ON (OFF)	Turn ON or OFF the automatic redraw of hits in the set views. The default is ON.
LIST	CLEAR	Clear the SETVIEW volume list. This is the list of volumes determined by the SETVIEW code to belong to the current view. The list may be treated as a single volume in the SETVIEW command.
NAME	ONLY (MANY)	Determine whether the volume name only will be drawn in the next set view, or the volume and all of its descendants. The default is MANY.
ORIGIN	Volume name followed by optional coordinates	Specifies the origin of the requested view. This is normally decided internally by the SETVIEW code. If no coordinates are given, the view origin is taken to be origin of the volume. The coordinates if present are taken to be in the reference system of the volume.
ORIGIN	FREE	Return the choice of the view origin to the SETVIEW routines.
PADS	ON (OFF)	Turn ON or OFF the automatic redraw of TPC pads in the set views. The default is OFF.
SEEN	Volume name followed by an optional integer to specify "all descendants"	Render a volume visible in the set views. (The GEANT SATT command accomplishes this for DRAW/DRAW commands but not for the set views, which depend on a view list particular to the SETVIEW code.) This command affects all of the subsequent views.
TRKS	ON (OFF)	Turn ON or OFF the automatic redraw of tracks in the set views. The default is OFF.

Option	Qualifier(s)	Meaning
UNSEEN	Volume name followed by integer (as above)	Render a volume invisible in the set views.

3 Commands

The commands in the sections that follow are listed alphabetical order. The "_" character is considered to follow the rest of the alphabet.

3.1 Commands that control the geometrical setup

The user is cautioned that it is necessary to issue the command CHANGE for most of the geometrical changes made by commands in this section to have an effect on the GEANT setup. The command CHANGE is documented in this section.

1. BEAM or /EOS_GEANT/RUN/BEAM RHGC_BEAM_ALPHA RHGC_BEAM_BETA

RHGC_BEAM_ALPHA	R	Beam alpha (degrees)
RHGC_BEAM_BETA	R	Beam beta (degrees)

Input the beam direction angles ALPHA and BETA. The handling of the beam direction in the GEANT simulation involves the specification of a LAB reference frame. The LAB frame is operationally defined in EOSGX as that frame whose origin is the event vertex and whose z axis coincides with the incident beam direction at the target. The y axis of the LAB frame is constrained to lie in a plane which is vertical in the CAVE system, and the x axis of the LAB frame completes a right-handed coordinate system.

The angles ALPHA and BETA are Euler angles specifying the rotation of the LAB frame in the CAVE system, starting with the axes of the two frames coincident. Angle ALPHA specifies a counterclockwise rotation of the LAB frame about the CAVE y axis (counterclockwise when looking down the y axis toward the CAVE origin). A positive value for ALPHA rotates the LAB z axis (and therefore the beam) westward relative to the CAVE z axis. Angle BETA specifies a counterclockwise rotation about the intermediate x' axis of a frame whose y' axis coincides with the CAVE y axis. A positive value for BETA has the effect of tilting the beam downwards in the CAVE system. For small BETA, ALPHA is approximately equal to the horizontal angular displacement of the beam.

The FFREAD card corresponding to this command has more arguments than the command. Thus the command

BEAM -2.5 -0.6

corresponds to the FFREAD card

BEAM $9 = -2.5$ $10 = -0.6$

(each gives a tilt of $0.6°$ upward, and approximately $2.5°$ eastward).

2. BFIELD or /EOS_GEANT/RUN/BFIELD C_FIELD [FIELD_MAG]

C_FIELD	C	Field option (EOS, HISS, CNST, EFFE, default = EOS)
FIELD_MAG	R	Value of the field in kilogauss (for CNST, EFFE)

This command loads a magnetic field map for GEANT. The format for the data file is assumed to be that of Doug Olson's field maps for HISS magnet. BFIELD EOS loads a field map assumed to be in Kilogauss and HTPC coordinates, and allows using the same data files as the track finding routines. BFIELD HISS loads a field map in Gauss and CAVE coordinates, which is more common for the HISS field maps. BFIELD CNST will use a routine to load a constant field (along $\hat{y}$), with the value determined by FIELD_MAG, if present, or 13kG. BFIELD EFFE will use a routine to load a constant field if $X^2 + Z^2 < R_{\mathrm{eff}}^2$, where R_{eff} is the effective edge radius. The value of the field (along $\hat{y}$) is determined by FIELD_MAG, if present, or 13kG is used. Note that EFFE is preferred over CNST since this field configuration reduces the number of spirals.

Only one map is present in memory, and it is possible to overwrite the field map used by the track finder if this command is given after an event has been track found, or the track finder can overwrite one of the GEANT field maps. The HISS option loads field maps that are incompatible with current track finders. The actual data files read are determined by the values of the logical names EOS_TRACKER_BFIELD (for option EOS) and HISS_GEANT_BFIELD (for option HISS) on the VAX. See the command FXYZ to find out what the field values are.

3. CHANGE or /EOS_GEANT/VOLUME/CHANGE

Drop current geometry/medium definitions (GIDROP) and load new ones. It is necessary to execute this command for any new geometrical definitions (e.g., as defined by TARGET) to be used by GEANT.

4. GEOMETRY or /EOS_GEANT/VOLUME/GEOMETRY X_CHAR

X_CHAR	C	Source keyword for geometry (FF or TT)

The geometry command determines the initial source for the detector positioning information. The positioning information for the detector layout is loaded from one of two possible sources: either the input from the FFREAD file, or the information in the ROTM table. The choice is made based on the value of the keyword in the GEOMETRY command, FF specifies the FFREAD file and TT specifies the table. The GEOMETRY command should be used at initialization time in one of the EOSGX logon macros.

Note: There are also GEANT commands using "GEOMETRY".

5. GVERT or /EOS_GEANT/RUN/GVERT CVRTX [VTX0 VTY0 VSIGMAX VSIGMAY]

CVRTX	C	Vertex option (SVTX, DVTX, default = SVTX)
VTX0	R	Mean x of vertex in TARG coordinates
VTY0	R	Mean y of vertex in TARG coordinates
VSIGMAX	R	Width of Gaussian in x
VSIGMAY	R	Width of Gaussian in y

This command specifies a choice of event vertex located within the target. The possible options are SVTX (single event vertex) and DVTX (distributed event vertex). The distributed vertex is currently required for the DATA run mode. Otherwise a single vertex is the default. In the case of a single event vertex, the vertex may be optionally smeared. The smearing is a Gaussian with widths of VSIGMAX and VSIGMAY transverse to the beam direction (centered at VTX0, VTY0 in TARG coordinates), and is uniform along the beam direction within the limits of the target thickness. Note that the command ORIGIN will override the definitions in GVERT.

6. MAXROW or /EOS_GEANT/EVENT/MAXROW MAX_TPC_ROWS

MAX_TPC_ROWS	I	Maximum number of rows

Maximum number of TPC pad rows. It is recommended that MAX_TPC_ROWS be 1 for FSIM mode so that the step size will be 1 mm regardless of pad row crossings. It is required that MAXROW be 128 in DSIM mode for the pad rows to be used correctly.

7. ORIGIN or /EOS_GEANT/EVENT/ORIGIN VNAM [VTX VTY VTZ]

VNAM	C	Volume name of local reference system
VTX	R	x coordinate of new vertex
VTY	R	y coordinate of new vertex
VTZ	R	z coordinate of new vertex

This commands inputs a fixed event vertex. If VNAM = MARS or CAVE the vertex given is assumed to be in the master (CAVE) coordinates. Otherwise the vertex origin is calculated in MARS from the given coordinates, which are taken to be in the local reference system of volume name VNAM. If no vertex coordinates are given, the vertex is taken to be the origin of the volume VNAM. Note that this is a fixed (non-smeared) vertex. The choice of vertex by the ORIGIN command overrides the choice of a (target) vertex by the GVERT command. To undo the ORIGIN command, use the command ORIGIN FREE which resets the vertex to that of the last valid GVERT command.

8. TARGET or /EOS_GEANT/VOLUME/TARGET TGNAM
[RHO_X TGX TGY TRGPOS(1) TRGPOS(2) TRGPOS(3)]

TGNAM	C	Name of target material (ALUMinum, CALCium, CARBon, COPPer, GOLD, NIOBium, OUT, default = GOLD)
RHO_X	R	Areal density in mg/cm^2 (default = 490)
TGX	R	Full width in x in cm (default = 9)
TGY	R	Full width in y in cm (default = 9)
TRGPOS(1)	R	x position in TPC coordinates (cm)
TRGPOS(2)	R	y position in TPC coordinates (cm)
TRGPOS(3)	R	z position in TPC coordinates (cm) (default = 0,0,0)

This command defines the target parameters. RHO_X is given in mg/cm^2 and the remaining dimensions are given in cm. Note that the position of the target is given in TPC coordinates. The command TARGET OUT will replace the current target with a 0.5 cm thickness of AIR.

3.2 Commands that enter particles to be tracked

1. KINE or /CONTROL/KINE IKINE [PKINE1 PKINE2 PKINE3 PKINE4 PKINE5 PKINE6 PKINE7 PKINE8 PKINE9 PKINE10]

IKINE	I	IKINE, (default = 1)
PKINE1	R	PKINE(1)
PKINE2	R	PKINE(2)
PKINE3	R	PKINE(3)
PKINE4	R	PKINE(4)
PKINE5	R	PKINE(5)
PKINE6	R	PKINE(6)
PKINE7	R	PKINE(7)
PKINE8	R	PKINE(8)
PKINE9	R	PKINE(9)
PKINE10	R	PKINE(10)

This is actually a GEANT command rather than a command in the EOSGX menu. It is included here because the interpretation of the variables in array PKINE is particular to EOSGX. Currently the KINE command is only used to specify the input kinematics and particle information in input mode INTG, generator 2. Other interpretations of the PKINE array can easily be made by the user for different values of IKINE. Currently in EOSGX:

IKINE	=	1
PKINE(1)	=	minimum polar angle in MARS (CAVE)
PKINE(2)	=	maximum polar angle in MARS
PKINE(3)	=	minimum azimuthal angle in MARS
PKINE(4)	=	maximum azimuthal angle in MARS
PKINE(5)	=	minimum kinetic energy per nucleon in GeV
PKINE(6)	=	maximum kinetic energy per nucleon in GeV
PKINE(9)	=	multiplicity
PKINE(8)	=	particle Z
PKINE(9)	=	particle A
PKINE(10)	=	not used

2. PKINE or /EOS_GEANT/EVENT/PKINE VOPT PKIN(1) PKIN(2) PKIN(3) PKCHID

VOPT	C	Volume name of reference system
PKIN(1)	R	First kinematic parameter
PKIN(2)	R	Second kinematic parameter
PKIN(3)	R	Third kinematic parameter
PKCHID	C	Type of kinematics (P, E), default = E

This command is used in various input modes to specify particle kinematics. The variables in array PKIN have two different interpretations, depending on the choice of the character ID PKCHID. If PKCHID = E then

PKIN(1)	=	kinetic energy per nucleon in GeV
PKIN(2)	=	polar angle theta in reference system VOPT
PKIN(3)	=	azimuthal angle phi in reference system VOPT

If PKCHID = P then

PKIN(1)	=	x component of momentum in reference system VOPT
PKIN(2)	=	y component of momentum in reference system VOPT
PKIN(3)	=	z component of momentum in reference system VOPT

VOPT may be the name of any known volume in the geometrical tree.

3.3 Commands that control the tracking of particles

1. TRACKING or /EOS_GEANT/RUN/TRACKING TRACKING_OPT

TRACKING_OPT	C	Tracking option

Control options for the GEANT tracking. Current items available under TRACKING are CONTROL, LIMIT, LIST, SET, and SPIRALS.

Option CONTROL allows the user control of the step size in FSIM mode, for volumes other than the TPC. The current step size outside the TPC in this mode is hardwired to be 2.0 cm. The setting of this variable from the KUIP command line is a possible future option. The form of the command is

TRACKING CONTROL ON (OFF)

The default is ON.

Option LIMIT limits tracking for those tracks passing out of the sides of the TPC. If this is turned ON, tracks above a certain minimum rigidity are stopped at the side walls. If this is turned OFF, particles will be tracked out to the CAVE boundary. The default is OFF. The command

TRACKING LIMIT ON 0.5

stops all particles below rigidity = 0.5 GeV/c at the side walls of the TPC.

Option LIST allows updating of the tracking list. This list determines whether particles are to be tracked or not. The default (in the absence of a defined list) is for all particles to be tracked. The command

TRACKING LIST IZ IA

enters a particle of charge IZ and mass IA into the list. See Section 2.3.1 for the convention used for IZ and IA in the case of particles not composed of baryons.

If the command TRACKING LIST is entered without other arguments, this has the effect of clearing the list. See option SET (described below) for the interpretation of the tracking list. The contents of the tracking list can be examined with the command INFO TRACKING (see Section 3.8).

Option SET determines whether the definition of the list of particles is to be "particles to be tracked" or "particles not to be tracked". The command

TRACKING SET INCLUDE

defines the tracking list as "particles to be tracked". Alternatively the command

TRACKING SET EXCLUDE

defines the tracking list as "particles not to be tracked".

Option SPIRALS determines the number of spirals a track may make within the active volume of the TPC. The command

TRACKING SPIRALS 3

allows a particle to complete three spirals in the TPC. The default number of spirals is 2.

3.4 Commands that control the data flow

1. RUNID or /EOS_GEANT/RUN/RUNID I_RUNID [X_CHAR]

I_RUNI	I	Run ID
X_CHAR	C	Character ID for early runs

This command inputs a TAS run number for run-dependent calibrations. Since the run number is undefined for simulation input, and many of the calibration routines require knowledge of the run number to determine the correct calibration constants, a run number must be entered by hand in a simulation in which accurate calibrations are necessary. This command allows input of an artificial run number into a common block. The subroutine EOSG_SET_RUNSTAT loads the value into the EVNT table. If X_CHAR is present then the run number is assumed to be from the first group of runs (the actual text is ignored).

2. RUN_MODE or /EOS_GEANT/RUN/RUN_MODE VOPT

VOPT	C	Run mode (NORM, DATA, DSIM, FSIM, default = NORM)

This command sets the mode of interaction between GEANT and TAS. In the DATA mode, GEANT takes data from the TAS tables and begins a GEANT event. In the

DSIM mode, GEANT processes simulated tracks and inserts hits into the HITS table alongside any real data hits that may be present. In FSIM mode GEANT processes simulated tracks and either writes a *.BDF file (named BDFILE.BDF), or loads the data into the DTS_RAW table. The command MAXROW should be used to change the number of effective TPC pad rows when switching between DSIM mode and FSIM mode (see Section 3.1). The NORM mode is the standard GEANT simulation mode. See also Sections 2.2 and 4.

3.5 Commands that control the display

The automatic drawing of the event display at the end of analysis is done by the TAS routine GEANT_DISPLAY. The TAS command:

SET/MODULE GEANT_DISPLAY ON (or OFF)

will enable (or disable) this display.

The commands are:

1. DDCUTS or /EOS_GEANT/DCONTROL/DDCUTS [VNAM DD_ITEM TAB_VAR DD_DLIM DD_ULIM]

VNAM	C	Name of major detector
DD_ITEM	C	Display item to which the cut applies (HITS, TRKS)
TAB_VAR	C	Table.variable expression
DD_DLIM	R	Lower limit of interval (default = -1000)
DD_ULIM	R	Upper limit of interval (default = 1000)

Define cuts on event display items. This command allows the definition of "permanent" cuts on items drawn in the event display (i.e., HITS or TRKS). The cuts are permanent in the sense that their definitions are kept in memory until explicitly deleted. The cuts are active when referenced by a command to draw hits or tracks (DDHITS or DDTRACKS).

Cuts may be specified for any "major detector" as defined in the event display — these have the GEANT names HTPC, MUSI, TOFN, MUFF, PLT1, PLT2. For example, the following defines a cut on the charge of hits in the TOF wall between the limits of 1 and 6 inclusive:

DDCUTS TOFN HITS TOF_ANL.TOF_IZ 1.0 6.0

Each cut is assigned a number when entered. The current cut definitions and assigned cut numbers are available with the command INFO CUTS.

Once cuts are entered there is no mechanism to delete them individually. Instead they are deleted as a group. For example, all currently defined cuts for the TPC may be deleted with the command

DDCUTS HTPC

without additional arguments. If the command DDCUTS is entered without arguments, all currently defined cuts are deleted.

2. DDHITS or /EOS_GEANT/DISPLAY/DDHITS [VNAM DD_COLO_HIT DD_MTYP_HIT DD_MSCF_HIT CUT1 CUT2 CUT3]

VNAM	C	Detector option (default = CAVE)
DD_COLO_HIT	I	Color of hit marker (default = 7)
DD_MTYP_HIT	I	Marker type for hits (default = 2)
DD_MSCF_HIT	R	Scale factor for marker (default = 0.75)
CUT1	I	First cut number (default = −10)
CUT2	I	Second cut number (default = −10)
CUT3	I	Third cut number (default = −10)

Display hits for the current event. With the use of this command, hit points in the major detectors may be displayed within the graphical representation of the CAVE. The color, marker type and size of the displayed space points may be specified by optional parameters.

In addition, cuts may be specified to display a subset of the total hits. These cuts must have been previously added to the list of permanently defined display cuts (see DDCUTS). A maximum of three cuts may be specified explicitly by cut number, and the drawn hits will be those satisfying all cuts. A cut number of zero is interpreted to mean "all defined cuts".

3. DDPADS or /EOS_GEANT/DISPLAY/DDPADS [ICOL]

ICOL	I	line color index (default = 4)

This command draws the pad plane in CAVE coordinates.

4. DDTRACKS or /EOS_GEANT/DISPLAY/DDTRACKS [VNAM DD_ICOLI CUT1 CUT2 CUT3 DD_STEPSI DD_RADIUS DD_IFLAG DD_MTYPE DD_PRINTF]

VNAM	C	Detector or option for which to draw tracks (default = CAVE)
DD_ICOLI	I	Color index (default = 0)
CUT1	I	First cut number (default = −10)
CUT2	I	Second cut number (default = −10)
CUT3	I	Third cut number (default = −10)
DD_STEPSI	R	Tracking step size (default = 2)
DD_RADIUS	R	Draw track out to radius (default = 800)
DD_IFLAG	I	Draw (symbol=1 or line=2)
DD_MTYPE	I	Symbol marker type (default = 0)
DD_PRINTF	R	Print flag (default = 0)

Display tracks for the current event. With the use of this command, tracks found in the TPC or MUSIC may be displayed within the graphical representation of the CAVE. The color, step size and other aspects of the drawn trajectory may be specified by optional parameters. The option CAVE (the default) will draw both the TPC tracks and the MUSIC tracks.

In addition, cuts may be specified to display a subset of the found tracks. These cuts must have been previously added to the list of permanently defined display cuts (see DDCUTS). A maximum of three cuts may be specified explicitly by cut number, and the drawn tracks will be those satisfying all cuts. A cut number of zero is interpreted to mean "all defined cuts."

The tracks will be labelled by the track number in the tables and the particle type (if known). Specifying DD_ICOLI less than zero will cause the numbering to be suppressed (and the absolute value of DD_ICOLI will be used for the color).

5. DD_HITS or /EOS_GEANT/DISPLAY/DD_HITS VNAM [TAB_VAR DD_DLIM DD_ULIM COLO_TMP MTYP_TMP MSCF_TMP]

VNAM	C	Detector for which to draw the hits
TAB_VAR	C	Table.variable expression
DD_DLIM	R	Lower limit of cut interval (default = −1000)
DD_ULIM	R	Upper limit of cut interval (default = 1000)
COLO_TMP	I	Color of hit marker (default = 0)
MTYP_TMP	I	Marker type for hits (default = 0)
MSCF_TMP	R	Scale factor for marker (default = 0)

Display hits subject to a single temporary cut. With the use of this command, a single cut may be imposed on the hit points displayed for a major detector. This cut is active only when the command is issued, and is deleted immediately afterwards. This differs from the treatment of "permanent" display cuts (see DDCUTS).

The cut is specified using a table.variable expression. As an example, a cut on 1-pad hits in the TPC would be specified as

DD_HITS HTPC HITS.NPADS 1.0 1.0

The color, size and marker type of the drawn space point may be optionally set with additional parameters.

6. DD_TRKS or /EOS_GEANT/DISPLAY/DD_TRKS VNAM [TAB_VAR DD_DLIM DD_ULIM DD_ICOLI_TMP]

VNAM	C	Detector or option for which to draw the tracks
TAB_VAR	C	Table.variable expression
DD_DLIM	R	Lower limit of cut interval (default = 0)
DD_ULIM	R	Upper limit of cut interval (default = 1000)
DD_ICOLI_TMP	I	Color index (default = 4)

Display tracks subject to a single temporary cut. With the use of this command, a single cut may be imposed on the found tracks displayed for a major detector. This cut is active only when the command is issued, and is deleted immediately afterwards. This differs from the treatment of "permanent" display cuts (see DDCUTS).

The cut is specified using a table.variable expression. As an example, a cut on tracks in the TPC with assigned charge between 1 and 3 (inclusive) would be specified as

DD_TRKS HTPC TRKS.IZ 1.0 3.0

The color of the drawn track may be optionally set with the additional parameter DD_ICOLI_TMP.

7. DRAWT or /EOS_GEANT/EVENT/DRAWT DT_OPT

DT_OPT	C	Option (ONLINE, OFFLINE or OFF)

Control storing/drawing of trajectories in the GEANT loop. ONLINE stores and draws the trajectories, OFFLINE stores the trajectories, and OFF will not store the trajectories.

8. OPTVIEW or /EOS_GEANT/DISPLAY/OPTVIEW CHOPT CHID [R_OPT1 R_OPT2 R_OPT3 VREF]

CHOPT	C	Name of option
CHID	C	Option qualifier
R_OPT1	R	First real parameter (default = 0)
R_OPT2	R	Second real parameter (default = 0)
R_OPT3	R	Third real parameter (default = 0)
VREF	C	Name of reference system (default = CAVE)

Special options for the set views.

Current options available under OPTVIEW are: COLOR, CLEAR, HITS, LIST, NAME, ORIGIN, PADS, SEEN, TRKS, and UNSEEN.

The various options have different qualifiers depending on the particular option selected. These are listed in Section 2.7.

9. REDRAW or /EOS_GEANT/DCONTROL/REDRAW

This command redraws the current view.

10. SETVIEW or /EOS_GEANT/DISPLAY/SETVIEW VNAM VOPT [VCHID THETA PHI PSI VREF]

VNAM	C	Volume name (default = CAVE)
VOPT	C	View name (default = AUTO)
VCHID	C	View mode ID
THETA	R	Polar angle of line of sight (default = 0.0)
PHI	R	Azimuthal angle of line of sight (default = 0.0)
PSI	R	Rotation of view in screen plane (default = 0.0)
VREF	C	Reference frame of line of sight (default = CAVE)

This command sets the drawing parameters for a view and draws the detector(s). Viewing options are AUTO, PERS, TOP, SIDE, BACK, FRON, ISOM(etric), which control the viewpoint. If the AUTO option is used in the SETVIEW command, control transfers to the graphics cursor. On VAXstations, this is usually controlled by mouse button MB1 (but on Sparcstations, MB2). Any main detector may be looked at in AUTO mode.

The most general command is

SETVIEW CAVE AUTO

In this case a top view of the CAVE appears, and input is requested. If the user clicks on a main detector, this detector is displayed in its PERSpective orientation. Clicking repeatedly causes the display to scroll through the different views of the selected detector. To exit, click on the lower left corner of the window. This returns the user to the top view of the CAVE, from which further selections may be made. If the user clicks on a portion of the screen not occupied by a detector, the display scrolls through the CAVE views. In a particular CAVE view it then becomes possible to click on a given detector in that view, and display that detector.

For any other choice, for example SETVIEW HTPC AUTO, clicking repeatedly causes a scroll through the views of the selected detector only.

VCHID identifies the mode of the view as 'G' global, 'L' local, or 'B' build. See also section 2.6.

3.6 Commands that control input and output of data

1. INPUT_MODE or /EOS_GEANT/RUN/INPUT_MODE VOPT

VOPT	C	Event input option (FILE, ONE, MANY, DATA, PRTS, default = FILE)

Select a mode of event input to GEANT. FILE reads from a FREESCO data file. ONE reads a single particle from the terminal (see Section 6.2 for an example). MANY is under development. In DATA input mode the GEANT kinematics banks are filled from the TRACKS table. The inclusion of MUSIC tracks is a possible future addition. This input mode is the only possible choice in DATA run mode (and is automatically selected when switching to DATA mode). The DATA input mode is also available in NORM mode, but not in the DSIM and FSIM run modes. PRTS will read data from the PARTS table (filled by the routine FREESCO).

2. LUN or /EOS_GEANT/RUN/LUN LUNIN [FILENAME CHID]

LUNIN	C	Logical unit number
FILENAME	C	Name of file (default =" ")
CHID	C	Open/close (default = "O")

This command selects the logical unit number to be used for GEANT event input in the FILE input mode of EOSGX.

If no filename is supplied, the unit is assumed to be one of those already opened for unformatted input. If a filename is supplied, the unit LUN will be assumed to be new, and will be opened for the new file. In either case, the logical unit LUN then becomes the current input unit for primary kinematics.

The optional character ID CHID determines the action taken on the logical unit. Default is "O" to open the file on the logical unit if not already open. If CHID = "C" the unit is closed.

At the start of the program EOSGX can open files given by the logical names:

Logical unit	File name
48	EOSG_INPUT_1
49	EOSG_INPUT_2
50	EOSG_INPUT_3

The control card GLUN (see Section 9.2) controls which, if any, of these files are opened when EOSGX is started. This card also determines which of the logical units is to be used by default. Currently, the file names are logicals that are defined in the startup file EOSGX_LOGICALS.COM (on the VAX).

3. RUN/REWIND or /EOS_GEANT/RUN/REWIND

   This command rewinds the current input file.

3.7 Commands that control the data being output

3.7.1 In run mode DSIM

1. HIT_JITTER or /EOSG_GEANT/RUN/HIT_JITTER X_CHAR

X_CHAR	C	Option (ON or OFF, default = ON)

   This command determines if the resolution effects are added to the hit positions returned from GEANT. If ON the x and y positions returned have a Gaussian distributed random number (with σ's determined by DX and DY in the routine GUOUT_SMDATA) added to them to mimic the resolution of the TPC. If a hit is outside the TPC's active volume after jittering, the hit position is moved to the closest point within the active volume.

2. HIT_MERGE or /EOSG_GEANT/RUN/HIT_MERGE X_CHAR

X_CHAR	C	Option (ON or OFF, default = ON)

   This command determines if the GEANT hits in the TPC are merged before being output. If ON then hits that are close to other hits are merged into a single large hit ("cluster"). If the size of the cluster is small enough, it is output as a single hit. Note that merging is done after hit "jittering" (if jittering is enabled).

3. HIT_SCALE or /EOSG_GEANT/RUN/HIT_SCALE X_CHAR

X_CHAR	C	Option (ON or OFF, default = ON)

   This command determines if the ΔE values are scaled before being output. The formula used to scale the values is $\Delta E_{out} = \Delta E \cdot \text{DEDXSCALE} + \text{DEDXOFFSET}$, and is used to match the Monte Carlo's energy calibration to the TPC's calibration, so that the normal PID routines may be used. Note that the scaling is applied after merging, and/or jittering (if either is enabled).

4. HIT_THRESHOLD or /EOSG_GEANT/RUN/HIT_THRESHOLD X_CHAR

X_CHAR	C	Option (ON or OFF, default = ON)

This command determines if a threshold is applied to the hits before being output. To mimic the TPC's threshold, a threshold on the ΔE of the particle (while above a pad row) can be applied. Note that the threshold cut is applied after merging, jittering, and/or scaling (if any of these are enabled).

3.8 Commands that query the setup

This class of commands includes commands that query the flags (both EOSGX flags and some GEANT flags), query the setup of the program, and the geometrical setup of the detectors.

1. FXYZ or /EOS_GEANT/EVENT/FXYZ XP YP ZP

XP	R	x coordinate in field region
YP	R	y coordinate in field region
ZP	R	z coordinate in field region

Find field components at a given point in MARS (MAster Reference System, i.e., CAVE coordinates).

2. INFO or /EOS_GEANT/EVENT/INFO CH_INFO [VNAM1 VNAM2]

CH_INFO	C	requested item
VNAM1	C	First volume name (default = CAVE)
VNAM2	C	Second volume name (default = CAVE)

This command outputs information about many aspects of EOSGX.

Currently available options are CUTS, DISPLAY, EOSG, FIELD, MEDIA, ONE, TRACKING, TRANS, TREE, Volume names, and XTR.

3. LOCATE or EOS_GEANT/VOLUME/LOCATE VRF1 VPT(1) VPT(2) VPT(3) VRF2 KNUMBV

VRF1	C	GEANT volume name of known reference system
VPT(1)	R	x coordinate in VRF1
VPT(2)	R	y coordinate in VRF1
VPT(3)	R	z coordinate in VRF1
VRF2	C	Reference system of requested coordinates
KNUMBV	I	vector of copy numbers to uniquely identify volume

This command locates a point known in a particular reference system VRF1 in another system VRF2. As an example, if VRF1 = MARS or CAVE the point is taken to already be in the MAster Reference System (named CAVE in EOSGX). Its new coordinates in the daughter reference system (DRS) of volume VRF2 are then calculated. An optional vector of copy numbers may also be given in order to uniquely identify a volume, in the case of multiple copies of that volume name existing in the geometrical tree. Therefore, the command

VOLUME/LOCATE PADS 0.0 0.0 0.0 HTPC 1 1

requests the coordinates of the center (0,0,0) of the first pad in the first pad row in reference system HTPC. The name PADS is not unique, so the vector (1,1) specifies the requested pad. Note that the command

VOLUME/LOCATE PADS 0.0 0.0 0.0 HTPC

requests the coordinates of the centers of ALL of the 15360 pads!

Note also that the GEANT numbering of detector elements is not necessarily equivalent in all cases to the numbering conventions used by the particular detector represented.

4. STATUS or /EOS_GEANT/EVENT/STATUS

   This command gives the status of flags (e.g., the HIT_JITTER flag) used for diagnostics and control.

3.9 Command name conflicts

GEANT uses the command GEANT/LISTS/SAVE and TAS uses the command TAS/SET/SAVE_TABLES, so that the command

SAVE

is ambiguous.

PAW uses the command NETWORK/PIAF/STATUS and TAS uses the command EOS_GEANT/EVENT/STATUS, so that the command

STATUS

is ambiguous.

3.10 GEANT mother volumes

These GEANT mother volumes define a detector system

Volume name	Detector system
CAVE	All of the detectors
HTPC	The TPC
MUSI	The MUSIC detector
TOFN	The TOF (small) wall
MUFF	The MUFFINS
PLT1	Pluto 1 (upstream)
PLT2	Pluto 2 (downstream)

3.11 Color Indices

Color Index	Color
1	black/white (opposite background)
2	red
3	green
4	dark blue
5	yellow
6	purple
7	sky blue
8	white/black (same as background)

The colors shown are the default values from PAW. Usually EOSGX is run with black and white reversed to give a black background.

3.12 Marker Types

Marker Index	Shape
1	Dot ($\cdot$)
2	Cross ($+$)
3	Asterisk ($*$)
4	Circle ($\circ$)
5	Times ($\times$)
6	Dot ($\cdot$)

4 TAS modules

This section lists all modules that are included in TEP_AM as a TAS module. The TAS command

SET/MODULE *module name* ON *or* OFF

will switch these modules on or off. The various run modes require that certain modules are ON and OFF, currently the necessary modules are switched ON and OFF by EOSGX. Some of the dependences are given below.

1. FREESCO – This module will generate a FREESCO event and load the information into the PARTS table.

2. GEANT_DISPLAY – This module transforms space points for display and displays the hits at the end of an event. Note that shutting OFF this module effectively disables hit displays from the command line since the hit buffers are not updated at the end of the event (however, some CPU is saved).

3. GEANT_FSIM_EVT, GEANT_DSIM_EVT, and GEANT_DATA_EVT – These modules initiate GEANT event loops within the TAS event loop. In FSIM mode the GEANT event loop is triggered at the start of the TAS loop, and the GEANT data can (optionally) be loaded into the DTS_RAW table for processing by the usual TAS routines. In DSIM mode the GEANT event loop is triggered after the TAS hit finder, and the data is added to the HITS table for processing by the track finder (and any routines that follow the track finder). In DATA mode the GEANT event loop is triggered at the end of the TAS event loop, the input data for GEANT being the tracks output by the TAS routines. Note that at most one of these modules should be ON at any given time.

4. GEANT_HIT_FINDER – In DSIM mode this module will add GEANT data to the HITS table after GEANT_DSIM_EVT triggers the GEANT event loop. If HIT_FINDER is OFF then only the GEANT data will be present in the HITS table. This module must be ON for DSIM mode to be useful.

5. GEANT_TRACK_SIM and GEANT_TRACK_DATA – In DSIM, FSIM and DATA modes these modules store the GEANT data in the TAS tables GEANT_TRACKS and GEANT_VERTEX_LIST. One of these modules must be ON for any comparisons between GEANT tracks and TAS tracks. At most one module should be ON at any one time. Technical note: These modules are two calls to the same routine (AM_GEANT_TRACK), made at different times in the analysis loop, and given different names in TEP_AM. The _SIM version is used in the simulation modes, and the _DATA version is used in the DATA mode.

6. TOF_SIM – This module is intended to simulate hits in the TOF wall. As of 28-Aug-1993, the TOF simulator jitters TOF change, X, and Y only.

7. TRK_EVALUATE – This module fills the tables TRACKDBG, GEANTDBG, and EVENTDBG from data in the GEANT_TRACKS, TRACKS, VERTEX_LIST and HITS tables. The CPU required is approximately equal to the track finder. For this module to run correctly the run mode must be DSIM.

8. TRK_SIM – This module is intended to simulate the perfect tracker. The HITS table is used to locate tracks with at least IGEANT_MIN_HITS (in table TRK_SIM_SW) and then if, the WRITE_TRACKS switch is set, the track data for these tracks is copied from the GEANT_TRACKS table to the TRACKS and TRACK_SUM tables, and HITS table is updated. The GEANT_TRACKS and GEANT_TRACK_SUM tables are updated (regardless of the value of WRITE_TRACKS).

5 Switches

This is a listing of switches used in the EOSGX routines (other than the GEANT switches, for those the user is referred to the GEANT manual [1]).

5.1 Switches in GEANT_SW

A list of the switch variables (in alphabetical order):

1. DEDXOFFSET – Offset added in HIT_SCALE to the dE/dx values after scaling. see also DEDXSCALE. The default value is $-7.770\text{E}{-7}$.

2. DEDXSCALE – Scale factor used in HIT_SCALE for the output dE/dx. The formula used is $dE/dx_{out} = \Delta E \cdot \text{DEDXSCALE} + \text{DEDXOFFSET}$; The default value is 2.396. Hits that would be scaled to negative values are ignored (see also HITTHRESHOLD).

3. HITTHRESHOLD – The hit threshold (in GeV/cm) for a hit to be considered good. Note that clusters of hits are merged and scaled *before* this cut is applied. Used in HIT_JITTER. The default value is $2.0\text{E}{-6}$ GeV. The hit scaling routine (if enabled) will cut all hits that would have been scaled to give negative values for dE/dx_{out} (these hits are included in the count of hits failing the threshold), thus if HITTHRESHOLD is negative, the codes will behave as though HITTHRESHOLD were 0.0.

4. ONEPADTH – The threshold below which hits will be treated as one pad hits. The uncertainty in x associated with the hit (DX) is given as XPADSIZE/$\sqrt{12}$ and the positions are rounded (in x) to a position of the form $x_{out} = \text{XPADSIZE} \cdot \left(N + \frac{1}{2}\right)$, where N is an integer, to reproduce some of the one pad hit effects. The treatment of y is the same as for the other hits.

5. ONEPADWIDTH – The uncertainty in dE/dx_{out} for the cut used to determine if a hit is a one pad hit or not, expressed as a fraction of the threshold value. The value used to determine if a hit is a one pad hit or not is DEDX_ARE+ONEPADWIDTH$\times$ ONEPADTH$\times$Gaussian, where GAUSSIAN is a Gaussian distributed random number with unit sigma, instead of DEDX_ARE. Thus, in the neighborhood of the cut,

there is a probability of being one pad or two pad rather than a just cutoff value. The default value is 0.20, which seems to give about the same distribution as the Ni + Cu data.

6. PRINTF – The print flag. If the flag is set to a negative, value the usual debugging output will be reduced. The default value is 0. Routines known to use this flag are TOF_SIM (suppresses output at -3).

7. SIGMA_A_SQ – Used in GUOUT_SMDATA to find the pad resolution function. The value is that of σ_A^2. The default value is 1.94E$-$3 cm^2.

8. SIGMA_B_SQ – Used in GUOUT_SMDATA to find the pad resolution function. The value is that of σ_B^2. The default value is 4.54E$-$6 cm.

9. SIGMA_C_SQ – Used in GUOUT_SMDATA to find the pad resolution function. The value is that of σ_C^2. The default value is 9.98E$-$3 cm^2.

10. SIGMA_Y – Used in GUOUT_SMDATA as the resolution in the $\hat{y}$ (drift) direction. The default value is 0.07 cm.

11. XMAX_CLUSTER_SIZE – Clusters of hits larger than this size will be cut rather than output as a merged hit. Note that the cluster size is taken to be in pad widths (see XPADSIZE) and is rounded up to the next integer. Used in HIT_MERGE. The default value is 10.0 pads.

12. XMERGE – If hit merging is enabled, hits closer together than XMERGE in the X direction and closer together then YMERGE in the Y direction are considered to be overlapping and to be merged. XMERGE and YMERGE are used to give a tall, thin rectangle of merged hits. XMERGE_B and YMERGE_B are use to generate a short, fat rectangle of merged hits. The two rectangles combine to give a cross-shaped region of merged hits as is seen in the data. See section 7.2. Used in HIT_MERGE. The default value is 0.8 cm.

13. XMERGE_B – If hit merging is enabled, hits closer together than XMERGE_B in the X direction and closer together then YMERGE_B in the Y direction are considered

to be overlapping and to be merged. Used in HIT_MERGE. The default value is 1.5 cm.

14. XPADSIZE – The dimensions of a pad in the X direction. Used to find number of pads that the cluster "hits." Used in HIT_MERGE. Also used in GUOUT_SMDATA, and HIT_JITTER as the pad width for the one pad hit calculations. The default value is 0.8 cm.

15. YMERGE – If hit merging is enabled, hits closer together than YMERGE in the Y direction and closer together than XMERGE in the X direction are considered to be overlapping. Used in HIT_MERGE. The default value is 2.8 cm.

16. YMERGE_B – If hit merging is enabled, hits closer together than YMERGE_B in the Y direction and closer together then XMERGE_B in the X direction are considered to be overlapping. Used in HIT_MERGE. The default value is 0.6 cm.

17. YPADPLANE – The location of the pad plane in TPC coordinates (the Y component). Used in GUOUT_SMDATA to find the drift distance given the Y coordinate of the hit in TPC coordinates. The default value is -34.6 cm.

5.2 Switches in PEAK_SWIT

The table PEAK_SWIT is normally used by the HIT_FINDER, but some of the switches are also used by the GEANT_HIT_FINDER as appropriate.

1. PRINT – The print flag. If the value is less than, or equal to -3, GEANT_HIT_FINDER will not print out the summary information about hit generation and usage.

5.3 Switches in TPCSIM_SW

1. AVAL – Avalanche flag. The values for the flag are: 0 for no avalanche fluctuations, 1 for Ploya fluctuations (CPU intensive), and 2 for Gaussian fluctuations (less CPU intensive). The default value is 2.

2. EXB – $\vec{E} \times \vec{B}$ flag. The values for the flag are: 0 to ignore $\vec{E} \times \vec{B}$ effects, and 1 to take the $\vec{E} \times \vec{B}$ effects from a lookup table. The default value is 1.

3. ISEED – Random number seed used in the avalanche fluctuations. The default value is 54321.

4. ITHRESH – ADC threshold in ADC counts. This should be set to match the experiment's hardware threshold (6 or 7). The default value is 7.

5. OUTP – Selects where to output the data from the slow simulator. The values of the flag are: 1 selects the *.BDF file, and 2 selects the DTS_RAW table. The default value is 2.

6. DIFN – RMS transverse diffusion in $\mu\mathrm{m}/\sqrt{\mathrm{cm}}$. The default value is 540.

7. DRFTV – Drift velocity in cm/μsec. The default value is 5.5.

8. GAIN – Gain. This factor sets the gas gain in the TPC. The default value is 5000.[3]

9. RNOISE – RMS noise in units of ADC channels. The default value is 2.2.

5.4 Switches in TRK_EVALUATE_SW

1. IGEANT_MIN_HIT – The minimum number of hits in the TPC for a GEANT track to be considered a possible-to-find track for the purposes of comparison to track finder tracks in TRK_EVALUATE. The default value is 10.

5.5 Switches in TRK_SIM_SW

1. IGEANT_MIN_HITS – The minimum number of hits in the TPC for a GEANT track to be considered a possible-to-find track for the purposes of writing to the TRACKS table by TRK_SIM. The default value is 10.

2. WRITE_TRACKS – A flag to control if the output from the TRK_SIM module is written into the TRACKS table. If 1 then the TRACKS table is overwritten; if 0 then the TRACKS table is ignored (however, the GEANT_TRACKS and GEANT_TRACK_SUM tables are updated). The default value is 1.

[3] This value has to be modified so that the simulated data matches the experimental data. A. D. Chacon, 24-May-1993.

3. DEDX_SCALE – Used to scale the dE/dX information from GEANT in TRK_SIM in a manner similar to the standard track finders (e.g., TRK). The value used has to be chosen to match the track finder is it desired to simulate.

4. PRINTF – A flag used to control printing of messages in TRK_SIM. Positive values enable more debugging output (until a maximum of 10). A value of zero gives the standard output. Negative values disable more error and warning messages until a value of -10, which disables all output. The default value is 0.

5.6 Switches in the codes

1. There are no such variables known at the present time.

6 Examples

6.1 Reading a Monte Carlo data file

This example shows how to read a FREESCO data file, and send the GEANT hits to the TAS data analysis routines:

1. Specify that you wish to read from a logical unit:

   LUN 48

   logical unit 48 is currently assigned in the file EOSGX_LOGICALS.COM when the logical name EOSG_INPUT_1 is defined.

2. Specify the running mode "simulated data" (see Section 2.2), where the GEANT hits are loaded into the HITS table, and can be processed by the TAS routines:

   RUN_MODE DSIM

3. Disable the HIT_FINDER module to disable processing of experimental data.

   SET/MODULE HIT_FINDER OFF

4. Enable and disable any other modules according to the desired data analysis.

5. At this point one can type:

   GO n

   where n is the number of events to process. At the present time[4] GO -1 will cause problems in that the HITS table is not cleared between events, and in this mode the GEANT hits are *added* to the HITS table.

6.2 Entering a single particle into GEANT

This example shows how to enter a single particle into GEANT.

1. Set EOSGX to the proper input mode:

[4] 7-Jul-1993

RUN_MODE DSIM

INPUT_MODE ONE

2. Enter the particle type (here a gold nucleus):

   PARTICLE 79 197

   In general, Z of the particle is the first argument, and A is the second. If A is -1, then Z is assumed to be the GEANT particle type, so that

   PARTICLE 9 -1

   will enter a π^-.

3. Enter the particle's kinematics using the command

   PKINE VOPT PKIN(1) PKIN(2) PKIN(3) PKCHID

   this command is documented in Section 3.2.

   To enter the particle's kinetic energy per nucleon use PKCHID = "E", PKIN(1) = kinetic energy per nucleon (GeV), PKIN(2) = θ (polar angle), and PKIN(3) = ϕ (azimuthal angle), in the volume specified by VOPT. Or to enter the components of the particle's momentum use PKCHID = "P", PKIN(1) = P_x, PKIN(2) = P_y, and PKIN(3) = P_z in the volume specified by VOPT.

   Examples:

   PKINE CAVE 0.1 0.0 0.0 E

   specifies a particle of 100 A MeV energy (per nucleon) along the Z axis, and

   PKINE CAVE 0.0 0.0 0.1 P

   specifies a particle of 100 MeV/c momentum along the Z axis.

4. Enter the particle's vertex (or origin):

   ORIGIN TARG

   for the vertex at the center of the target. In general

ORIGIN CAVE −300.0 0.0 40.0

will put the vertex at CAVE coordinates −300.0, 0.0, 40.0.

5. Then process the particle – either enter:

   TRIG

   to start the GEANT processing (in NORM mode, see Section 2.2).

   Or, to process the particle, and then send the simulated data to the track finder use:

   GO 1

   In this mode it is assumed that the proper TAS commands have been used to set the proper run mode, disable the TAS hit finder (as required), disable the GEANT hit finder (as required), and enable the "event loop" without a data file, as was demonstrated in the example of reading a Monte Carlo data file.

7 Algorithms

7.1 Hit resolution

The formula used for the pad resolution function was taken from STAR note # 34 [4]. The resolution function is

$$\sigma_x^2 = \sigma_A^2 + \sigma_B^2 \lambda \, \exp \gamma\lambda \, \sec\alpha + \sigma_C^2 \, \exp\gamma\lambda \, \cos\alpha \, \tan^2\alpha \ , \tag{2}$$

where λ is the drift distance (taken in meters), α is the crossing angle, and $\exp\gamma\lambda$ is the inverse of an attenuation factor. The recommended values are $\sigma_A = 440\mu$m, $\sigma_B = 213\mu$m (for λ in meters), and $\sigma_C = 999\mu$m, and γ is taken to be zero. The code uses the values of σ^2 to avoid calculating a lot of squares. Also, GEANT uses centimeters rather than meters, then $\sigma_B^2 = 4.58 \cdot 10^{-6}$ cm.

The resolution in the y direction is taken as a constant, σ_y. The recommended value is $\sigma_y = 0.07$ cm.

For one pad hits (those which are above the value of HITTHRESHOLD and below the value of ONEPADTH) the position is rounded to the center of the nearest pad row by making the value of x of the form $x_{out} = \text{XPADSIZE} \cdot \left(N + \frac{1}{2}\right)$, where N is an integer, and XPADSIZE is the width (in x) of a pad. The value of the resolution returned (DX) is XPADSIZE/$\sqrt{12}$.

7.2 Hit merging

The hit merging algorithm uses only the space point information for the hits. It was decided, in the interest of speed, to not use the pad structure. A cluster of hits is found by numbering all the hits in a pad row. Then on each pass through every hit is compared to every other hit, if the two hits are within the merging distances ($\Delta x <$ XMERGE and $\Delta y <$ YMERGE, or $\Delta x <$ XMERGE_B and $\Delta y <$ YMERGE_B) the hit numbers are compared, if they are different, then the larger hit number is set to the lower hit number. After sufficient passes, the hit number of the lowest hit in a cluster will propagate throughout the entire cluster and identify hits in the cluster. Sufficient passes is the number of hits in the largest cluster (in the pad row) minus one, since on each pass at least one number must change, and the number of the lowest hit will not be changed. The program uses as an upper bound the number of hits in the pad row, and uses a flag to watch for the case that no hit numbers were changed (which indicates the end of the cluster finding). Note that this algorithm allows clusters of arbitrary size and shape. The size of the cluster (NPADS = Δx/XPADSIZE) is compared to a cut (XMAX_CLUSTER_SIZE) and it is not output if it is too large.

The hit that is the result of the merging has the following parameters associated with it:

1. GEANTID, the GEANTID of the hit that contributed the most energy to the cluster.
2. TRACKID, this is not used.
3. PADROW, all the pad rows in a cluster are the same so the first is used.
4. NPADS, this is set to the cluster size in x divided by the pad width.
5. SATF, this is not used.
6. DECON, this is not used.
7. X, this is set to the weighted (by DEDX_ARE) average of the X's in the cluster.
8. Y, this is set to the weighted (by DEDX_ARE) average of the Y's in the cluster.
9. Z, this is set to the weighted (by DEDX_ARE) average of the Z's in the cluster.
10. DX, this is set to the sum, in quadrature, of the variance of the hits in the cluster, and the (weighted sum of the) DX's of the hits in the cluster
11. DY, this is set to the sum, in quadrature, of the variance of the hits in the cluster, and the (weighted sum of the) DY's of the hits in the cluster
12. SIGMA_PRF, this is unused.
13. DEDX_AMPL, this is set to the largest DEDX_AMPL of all the hits in the cluster.
14. DEDX_ARE, this is set to the sum of the DEDX_ARE in the cluster.
15. ALPHA, this is not used.

16. LAMBDA, this is not used.

17. XDEV_FIT, this is not used.

18. YDEV_FIT, this is not used.

These values overwrite the first hit in the cluster. The remaining hits are flagged "invalid." The sum that is taken for the DX is (taking x_i as the X for the hits and w_i as the DEDX_ARE for the hits):

$$\mathrm{DX}^2 = \frac{\sum_{i=1}^{N} w_i x_i^2}{\sum_{i=1}^{N} w_i} - \left[\frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i}\right]^2 + \frac{\sum_{i=1}^{N} w_i \mathrm{DX}_i^2}{\sum_{i=1}^{N} w_i} \tag{3}$$

and the sum for DY is similar. Note that the first two terms are just $\sum w_i (x_i - x_{avg})^2 / \sum w_i$ which is what the hit finder will calculate if it is unable to resolve the hits.

7.3 Track evaluation

There are two sections to the code. The first section assumes that the track finder is correct and then searches for the best GEANT match, the second section of code assumes that GEANT is correct and searches for the best track finder match. When searching for tracks the routine sorts the hits according to the track number and then looks for indices such that the track number changes. The "best" match is found by sorting the hits within a track, and the ID that gave the most hits to this track is then considered the best match. The output is in the tables TRACKDBG (which assumes the track finder is correct and looks for GEANT matches) and GEANTDBG (which assumes the reverse).

In the event that a GEANT track has no corresponding track finder track the TRACKID is returned as zero. In the event that a TRACKID or GEANTID in the HITS table has no corresponding entry in the TRACKS table or the GEANT_TRACKS table, the TRACKID or GEANTID is returned, but the other values in the table corresponding to the TRACKID or GEANTID are set to invalid values.

8 TAS table use

8.1 EVENTDBG table

This table does comparisons of event features, and is filled in by the EOSGX routine TRK_EVALUATE. The data for this table are taken from the TRACKDBG, GEANTDBG, VERTEX_LIST and GEANT_VERTEX_LIST tables.

MULTIGEN	Multiplicity in the generated event
MULTIREC	Multiplicity in the reconstructed event
VERGEN(3)	The main vertex given to GEANT
VERREC(3)	The main vertex as reconstructed

The multiplicity of the generated event is subject to the cut given by IGEANT_MIN_HITS (see Section 5.4).

8.2 GEANTDBG table

This table assumes that GEANT tracks are correct and finds the best track finder match.

TRACKID	Track finder's ID for the best matching track
GEANTID	GEANT's ID for this track
PREC(3)	Rigidity of the track TRACKID
PGEN(3)	Rigidity of the track GEANTID
PTOT	The magnitude of PGEN
IAREC	A of the track TRACKID
IZREC	Z of the track GEANTID
IAGEN	A of the track GEANTID
IZGEN	Z of the track GEANTID
X0REC(3)	Location of the vertex from the track finder
X0GEN(3)	location of the vertex given to GEANT
NGEN	Number of hits in the GEANT track
NREC	Number of hits in TRACKID from GEANTID
NSEG	Number of track finder tracks using this GEANTID
FRAC	Fraction best match is of the total hits (NREC/NGEN)
IGPID	Geant particle type of GEANTID
NUSE	The number of TRACKID's that use this GEANTID as the best match

Note that if none of the hits in a GEANT track were used then TRACKID is returned as zero.

8.3 GEANT_TRACKS table

This table is filled in by GEANT_TRACK and TRK_SIM for each GEANT track as:

IA_PRT	GEANT's A for the particle
IZ_PRT	GEANT's Z for the particle
IPART	GEANT's particle ID
VRT(3)	Initial vertex location for this track
P(3)	Initial $\vec{P}$ for the particle
P_F(3)	$\vec{P}$ for the particle at the end of tracking
DEDX_TPC	Average dE/dx in the TPC
X_F(3)	Final position
GEANTID	GEANT track ID (GEANT's ITRA)
NUSE	The number of TRACKID's that use this GEANTID as the best match
IACC_TPC	Acceptance status in the TPC
IACC_TOF	Acceptance status in the TOF wall
IACC_MUSI	Acceptance status in the MUSIC chamber
DNN_TPC	Distance to the nearest neighbor in the TPC
DNN_TOF	Distance to the nearest neighbor in the TOF wall
DNN_MUSI	Distance to the nearest neighbor in the MUSIC chamber

Positions and momenta are given in the TPC's coordinate system. The variable NUSE used by TRK_EVALUATE to simplify the coding of TRK_EVALUATE.

8.4 GEANT_TRACK_SUM table

This table is filled in by GEANT_TRACK and TRK_SIM for each GEANT track as:

G_MULTI	GEANT changed particle multiplicity
G_NALPHA	Number of ^{4}He
G_NDEUT	Number of ^{2}H
G_NHE3	Number of ^{3}He
G_NPIM	Number of π^-
G_NPIP	Number of π^+
G_NPROT	Number of protons
G_NTRIT	Number of ^{3}He
G_ZSUM	Sum of Z
G_MULTIG6	Number of $z > 6$ particles
G_NLI6	Number of ^{6}Li
G_NLI7	Number of ^{7}Li
G_NCHPRI	Number of charged primaries
G_NCHSEC	Number of charged secondaries
G_NNEUT	Number of neutrals
G_ACCTPC	Number of particles accepted in the TPC
G_ACCTOF	Number of particles accepted in the TOF wall
G_ACCMUSI	Number of particles accepted in the MUSIC chamber
G_EEXCT	remnant excitation energy
G_SUML7	Sum of Z for $Z < 7$

8.5 GEANT_VERTEX_LIST table

This table is filled in by GEANT_TRACK for each GEANT event as:

VTX(3)	GEANT interaction vertex

Positions are given in the TPC's coordinate system. In DSIM and FSIM modes the first (and currently the only) vertex is the initial interaction vertex.

8.6 HITS table

This table is filled in by EOSGX (routine AM_GEANT_HITFINDER) for each hit as:

GEANTID	ITRA	Geant track ID
TRACKID	0.0	Track finder's track ID
PADROW		Pad row number from simulation
NPADS	1.0	Number of pads hit
SATF	0.0	Nedge
DECON	0.0	Decon
X		x from the simulation (TPC coordinates)
Y		y from the simulation (TPC coordinates)
Z		z from the simulation (TPC coordinates)
DX		σ_x from the simulation
DY		σ_y estimate
SIGMA_PRF	1.0	Sigma_prf
DEDX_AMPL		ΔE for the particle above this pad row
DEDX_AREA		ΔE for the particle above this pad row
ALPHA	0.0	Alpha
LAMBDA	0.0	Lambda
XDEV_FIT	0.0	Xdev_fit
YDEV_FIT	0.0	Ydev_fit

When the hits are merged the output is modified slightly, see Section 7.2.

8.7 PARTS table

This table is filled in by FREESCO (routine FREESCO) for each event as:

IA	A of the particle
IG	GEANT particle ID number
IP	Particle number from generator
IZ	Z of particle
MASS	Mass of the particle (Real variable)
P4(4)	The four-momentum of the particle (lab. frame)

8.8 TRACKDBG table

The TRACKDBG table, and the closely related GEANTDBG table are filled in by the EOSGX routine TRK_EVALUATE. The data for this table are taken from the TRACKS and GEANT_TRACKS tables, using the information in the HITS table to decide which track finder and GEANT tracks make the best match.

This table assumes that track finder tracks are correct and finds the best GEANT match.

TRACKID	Track finder's ID for this track
GEANTID	GEANT's ID for the best matching track
PREC(3)	Rigidity of the track TRACKID
PGEN(3)	Rigidity of the track GEANTID
PTOT	The magnitude of PGEN
IAREC	A of the track TRACKID
IZREC	Z of the track GEANTID
IAGEN	A of the track GEANTID
IZGEN	Z of the track GEANTID
X0REC(3)	Location of this track's closest approach or an error
X0GEN(3)	Location of the vertex given to GEANT
DELX(3)	The change in x from the first hit to the last hit
NGEN	Number of hits from GEANTID used by TRACKID
NREC	Number of hits in TRACKID
NSEG	Number of GEANT tracks in this TRACKID
FRAC	Fraction best match is of the total hits (NGEN/NREC)
IGPID	Geant particle type of GEANTID
DCA	DCA from the TRACKS table (distance of closest approach)
CHI	CHI2 from the tracks table (Currently χ^2/NDF)

9 Initialization

9.1 Files used

A number of KUMAC files are used to initialize EOSGX, most of these are called automatically from within PAW.

Some of these files are:

1. PAWLOGON.KUMAC – Called automatically from within PAW/TAS/GEANT.
2. TASLOGON.KUMAC – Called automatically from within PAW/TAS/GEANT.
3. GEANTLOGON.KUMAC – Called automatically from within PAW/TAS/GEANT.
4. GEANT's FFREAD file – This file can define the geometry of the detector, and sets GEANT's flags.

9.2 FFREAD card definitions

GEANT has the provision to set a number of its parameters from a file using a format-free read. The lines in the file are known as "cards." A number of cards have been defined to control EOSGX and these are documented here. Some additional cards, not defined here, are used to set GEANT parameters and are documented in the GEANT manual[1] (see also the section on FFKEY). Note that the geometry can also be specified through the use of TAS tables. The command GEOMETRY is used to control which initialization is used (see section 3.1).

FFREAD user cards defined in EOSGX:

1. BEAM

   Gives the angle of the beam. See the BEAM command for the definition.

2. BMLN [X] [Y]

   Locates the center of the beam line in CAVE coordinates. X and Y are in centimeters.

3. GLUN [LUN_INPUT] [IF_INPT1] [IF_INPT2] [IF_INPT3]

Assign logical unit numbers for GEANT input from an unformatted file.

LUN_INPUT – Unit number to be used as default (48, 49 or 50)

IF_INPT1 – flag to open unit 48 (INPUT_INPT1)

IF_INPT2 – flag to open unit 49 (INPUT_INPT2)

IF_INPT3 – flag to open unit 50 (INPUT_INPT3)

The unit number can be changed later with the LUN command at the TAS prompt. The actual file name to be opened is determined by the value of the logical names INPUT_INPT1 to INPUT_INPT3.

4. MAXR [MAX_TPC_ROWS]

   Maximum number of row divisions in the TPC. In DSIM mode this should be set to 128. In FSIM mode the value should be 1. See also the command MAXROW.

5. MUFF [X] [Y] [Z] [ROT] [THETA]

6. MUSI [X] [Y] [Z] [ROT] [THETA]

7. TOFN [X] [Y] [Z] [ROT] [THETA]

   These cards define the position of the MUFFins, MUSIc and TOF detectors. In each case X, Y, Z are the center of the detector in CAVE coordinates (cm). ROT is the angle of rotation of the detector in a horizontal plane about the CAVE center; THETA is the polar angle of the detector relative to the CAVE Z axis for a rotation in the horizontal plane about the detector's own center. Both angles are in degrees.

   The easiest scheme for positioning a detector with these cards is to set ROT=0 and determine the detector's angle relative to the beam with THETA.

References

[1] The GEANT team, *The GEANT users guide*, The CERN computing center. This is distributed as an electronic document, so there is no publisher or author in the usual sense.

[2] The GEANT team, GEANT — *Dectector Description and Simulation Tool*, The CERN computing center. CERN Program Library Long Writeup W5013.

[3] M. Goossens Ed., *PAW – Physics Analysis Workstation*, The CERN computing center write up Q121 (1989).

[4] P. Jones and G. Rai, *The STAR TPC pad resolution function*, STAR note # 34.