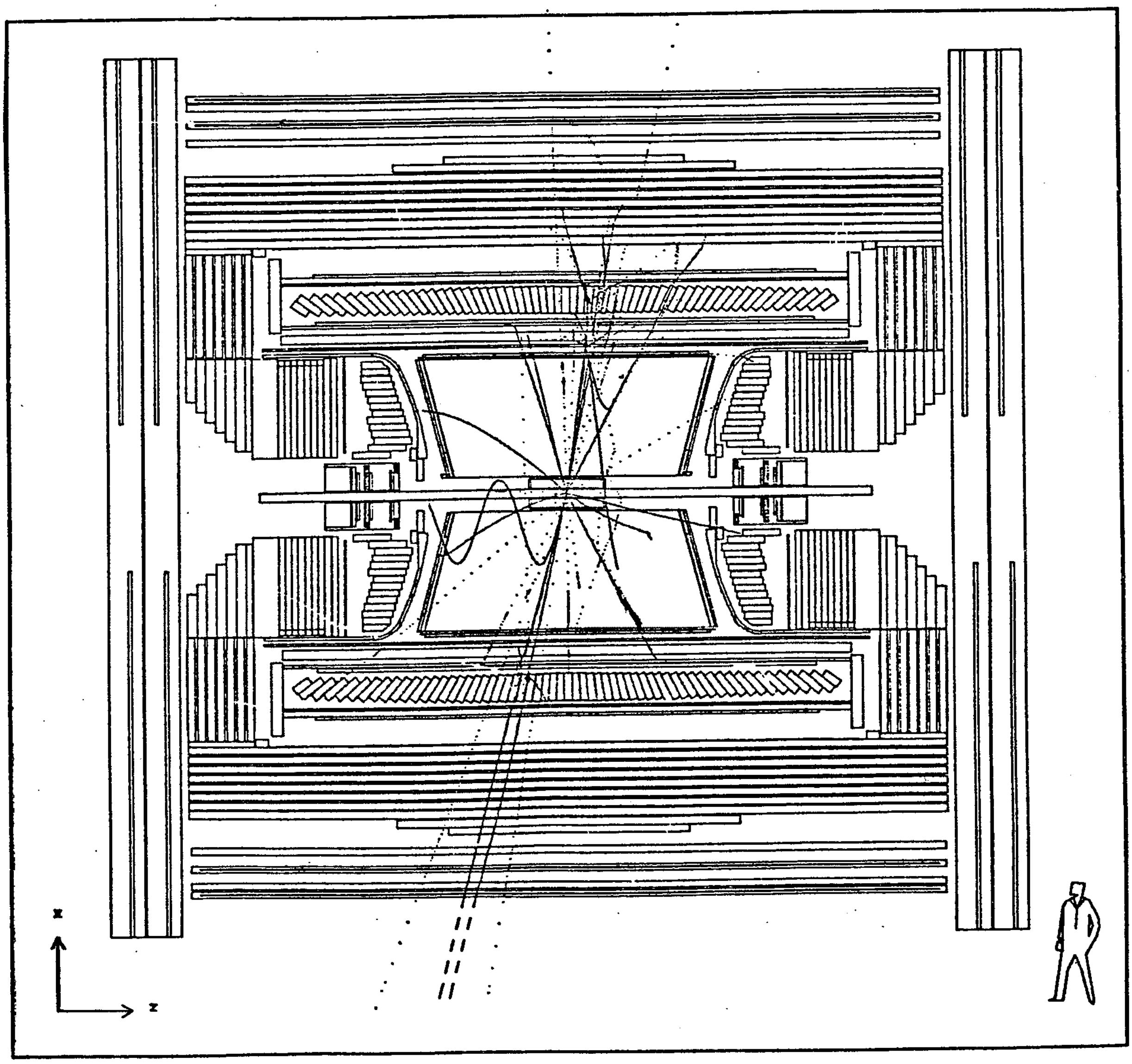
Gabriella

DATA HANDLING DIVISION
DD/EE/84-1
September 1987

Authors: R. Brun
F. Bruyant
M. Maire
A.C. McPherson
P. Zanarini

GEANT3



DOC-11551

GEANT 3.11 USER'S GUIDE AAAA 000

Author(s) : F.G.de Bilio
Origin : GEANT3

Submitted: 01.10.84
Revised: 09.09.87

Table of Contents

*) Routines revised

+) New routines

Author(s) : Undersigned
Origin : GEANT3*

Submitted: 01.10.84
Revised: 17.08.87

Foreword

As the scale and complexity of High Energy Physics experiments increase, simulation studies require more and more care and become essential

to design and optimize the detectors,
to develop and test the reconstruction and analysis programs, and
to interpret the experimental data.

GEANT3 is a system of detector description and simulation tools which should aid the physicists for such studies.

The GEANT3 system can be obtained from CERN as three Patchy PAM files exportable in Ceta format, GEANT3, GEANG3 and GEANX3. The material is currently tested on Apollo, IBM and VAX machines. The Pam files GEANT3 and GEANG3 contain most of the basic code. The Pam GEANX contains mainly the installation procedures and a few examples of application programs which may help non-familiar users to start with their own applications. It also contains a Patch of proposed corrections for the bugs found in GEANT3 and GEANG3.

General informations concerning GEANT3, for example access to the source code, the list of bugs and the proposed corrections, the context of utilisation on the CERN machines, the status of some application programs, the acquisition of documentation, etc., are kept up-to-date at CERN.

The first version of GEANT was written in 1974 as a bare framework which initially emphasized tracking of either one or a few particles per event through relatively simple detectors. The system has been developed with some continuity over the years.[1]

In GEANT3 new structural concepts have been introduced: The description of the geometrical setup and of the sensitive detectors is new, the overall strategy for the tracking has been rethought, most of the physics routines have been upgraded and, for the hadronic processes, an interface with the program GHEISHA[2] is now available and may be used instead of the default package derived from the program TATINA[3]

From version 3.10 up GEANT3 is using the Memory Manager ZEBRA.[4]

*) All quoted contributors.
We regret to mention here the accidental death of our collaborator and friend W. Gebel.

[1] GEANT User's Guide, CERN DD/78/2
GEANT2 User's Guide, CERN DD/EE/83-1

[2] Program GHEISHA, H. Fesefeldt, AACHEN 3rd Inst.

[3] Program TATINA, T. Baroncelli, INFN – Roma.

The current version may differ significantly from the previous one. Some of the modifications may lead to backward incompatibilities. The user is therefore invited to read carefully the Patch HISTORY of the current GEANT3 Pam file where all changes are described in detail.

The undersigned people responsible for maintaining GEANT3 are greatly indebted to many users whose contributions to application programs have been most useful in the initial debugging phase and have often led to improvements of the basic concepts. Their names are quoted in the documentation, hopefully with no omission.

Special mention should be made here of the work done by the following collaborators :

L. Urban (Budapest), upgrade of electromagnetic processes
F. Carminati (CERN-DD), interface with the program GHEISHA
G.N. Patrick (RAL), basic electromagnetic processes
D. Ward (Cambridge), many useful suggestions
J. Allison (Manchester), interface with the program CASCADE84, not mantained anymore.

Any reader who is not familiar with GEANT should first have a glance at the Short Write-up which can be obtained automatically (or manually) by selecting the notes numbered 001 to 009 in all sections of the User's Guide.

Many colleagues have kindly accepted to read preliminary draft copies of this write-up. Their remarks and corrections have been most appreciated. We would like to acknowledge especially the suggestions made by :

R. Hemingway (Carleton U., Ottawa)
H. Newman (Caltech)
S. O'Neale and F. Ranjard (CERN/EP).

Despite this, the documentation is still incomplete and far from perfect. We accept full responsibility for its present status.

We express our thanks to F.G. de Bilio for his most valuable contribution to the preparation of this document, which was produced with the text-editing program SCRIPT and SGML.

R. Brun (CERN/DD)
F. Bruyant (CERN/EP)
M. Maire (LAPP, Annecy)
A.C. McPherson (Carleton U., Ottawa)
P. Zanarini (CERN/DD)

[4] ZEBRA User Guide, CERN Program Library Q 100

Author(s) : F. Bruyant
Origin : GEANT3

Submitted: 01.10.84
Revised: 16.11.87

Introduction to the User's Guide

The present documentation is divided into ten sections which underline the structure of GEANT3 and its major functions. Each section is identified by a *keyword*, usually related to the name of the Patch where the relevant material should be found in the GEANT3 Pam files.

The sections are filed in alphabetic order :

BASE	GEANT3 framework and user interfaces (|to be read first|)
CONS	Particles,Materials and Tracking medium parameters
DRAW	The Drawing package
GEOM	The Geometry package
HITS	The Detector Response package
IOPA	The I/O package
KINE	Event generators and kinematic structures
PHYS	Physics processes
TRAK	The Tracking package
XINT	Interactive tools based on command processor ZCEDEX.

Within each section, the principal system functions or the details of subroutines are described in a series of 'papers' numbered from 001 to 999. The numbers have been allocated so as to give to the text some continuity. In the papers dedicated to the description of subroutines the authorship usually refers to the code and the source of the algorithms or contributions to the ideas are acknowledged under the item "Origin". In all other papers the authorship refers to the documentation itself and people responsible for the concepts or structures discussed there are quoted under the item "Origin".

Subroutines which are not necessary to understand the program flow and which are unlikely to be called directly by the user have been deliberately ignored.

The notation ['section'] is used whenever additional information can be found in the quoted section. In the description of subroutine calling sequences the arguments used both on input and on output are preceded by a * and the output arguments are followed by a * .

For convenience, two more sections have been opened :
the section *AAAA*, for general introductory information at the beginning, and
the section *ZZZZ*, for various appendices and indexed lists at the end

A table of contents is available in AAAA 000.

To ease access to this documentation an index is proposed in ZZZZ 999. It gives in alphabetic order the names of all documented GEANT3 subroutines with references to the appropriate write-up(s).

The 'loose-leaf' format of this User's Guide will permit to circulate partial updates or additions more frequently.

A short write-up of GEANT3 can be obtained by collecting the papers numbered 001 to 009 in all sections.

Author(s) : F. Bruyant
Origin : GEANT3

Submitted: 01.10.84
Revised: 30.05.86

Introduction to GEANT3

1. GEANT3 applications

The principal applications of GEANT3 are:

- the tracking of particles through an experimental setup for acceptance studies or simulation of detector response, and
- the graphical representation of the setup and of the particle trajectories.

It is of course desirable and very instructive to combine the two interactively since the observation of what happens to a particle during the tracking makes the debugging easier and may underline the weaknesses of the setup.

In view of these applications, the GEANT3 system allows:

- to describe an experimental setup in a rather efficient and simple way. The setup is represented by a structure of geometrical VOLUMEs. Each volume is given a 'MEDIUM' number by the user. Different volumes may have the same medium number [GEOM]. A medium is defined by a set of parameters, the so-called TRACKING MEDIUM parameters, which include reference to the MATERIAL filling the volume [CONS].
- to generate simulated events from standard Monte Carlo generators [KINE].
- to control the transport of particles through the various regions of the setup, taking into account the geometrical volume boundaries and all physical effects due to the nature of the particles themselves, to their interactions with the matter and to the magnetic field [TRAK, PHYS].
- to record the elements of the particle trajectories and the response from the sensitive detectors [TRAK,HITS],
- to visualize either interactively or remotely the detectors and the particle trajectories [DRAW, XINT].

Part of the subroutines available in GEANT3 are integrated into program segments prepared for the execution of some of these tasks.

The program segments may contain 'dummy' and 'default' user subroutines called whenever application dependent actions are expected.

The remaining subroutines are tools either to perform simple functions (control print, debug, I/O, etc.) or to implement the non trivial operations required for most of the applications (description of the geometrical setup, handling of detector responses, etc.).

It is the responsibility of the user to assemble the appropriate program segments and tools into an executable program, to code the relevant user subroutines, to provide the data describing the experimental environment and to submit the appropriate data cards which control the execution of the program. The section BASE of the User's Guide gives the information necessary to understand how to do the job.

Note: As a general convention the names of the dummy or default user subroutines have GU as first two letters and are printed in ***bold characters***.

2. Event simulation framework

The framework for event simulation is described in the following paragraphs to familiarize the reader with the areas where user interventions are expected.

At the same time, the GEANT3 data structures are introduced. This last point is important as the coding to be provided by the user most often consists of filling data structures, or extracting information from them, or saving them on output, making use of standard routines available in the system.

A main program has to be provided by the user [BASE 100]. It allocates the dynamic memory for ZEBRA and HBOOK and gives control to the three phases of the run:

- Initialisation
- Event processing
- Termination.

3. Initialisation

The initialisation phase is under the control of the user [BASE 100]. It consists of the following steps, most of them performed through calls to standard GEANT3 subroutines:

- GINIT, to initialize the GEANT3 labelled common blocks with default options which the user should be aware of [BASE 030, 110].
- GFFGO, to read 'free format' data cards with the possibility of overriding the default options [BASE 040, 110].
- GZINIT, to initialize the dynamic core divisions, the link areas and the Run header bank JRUNG [BASE 110].
- GDINIT, to initialize the drawing package [DRAW].
- GPART and auxiliaries, to generate the data structure JPART describing the standard particle properties [CONS].
- GMATE and auxiliaries, to generate the data structure JMATE describing the characteristics of the most commonly used MATERIALs [CONS].
- User code,
  – to define the geometry of the different components of the experimental setup [GEOM] and the tracking medium parameters [CONS,TRAK], and to generate the corresponding data structures JROTM, JVOLUM and JTMED.
  – to specify which elements of the geometrical setup should be considered as 'sensitive detectors', giving a 'response' when hit by a particle [HITS].
- GBHSTA, to book standard GEANT3 histograms as required by the user with the data card HSTA [BASE 040, 110].
- GPHYSI, to compute energy loss and cross section tables and to store them in the data structure JMATE [CONS,PHYS].

4. Event processing

The processing phase is triggered by a call to the subroutine GRUN which, for each event to be processed, gives control to the subroutines:

- GTRIGI, to initialize event processing and to create the Event header bank JHEAD.
- GTRIG, to process one event.
- GTRIGC, to clean up the portion of dynamic core used by the event

and checks that enough time is left for the next event [BASE 200].

The main steps of GTRIG consist of calls to the following user routines:

- ***GUKINE*** generates the data structures JVERTX and JKINE describing the kinematics of the current event on input [KINE], or reads them [IOPA].

- ***GUTREV*** calls GTREVE which controls the tracking for the whole event [TRAK]. Each particle is tracked in turn and when a sensitive detector is hit, the user may store any useful information in the data structure JHITS [HITS].
  Before going to the next particle any secondary products generated by a given particle, and stored by the user in the temporary data structure JSTAK, are processed in the same way. Simultaneously, the data structure JXYZ, containing the coordinates of space points along the tracks for the whole event, can be filled by the user [TRAK].

- ***GUDIGI*** simulates the detector responses for the whole event, making use of the information previously recorded in the data structure JHITS, and stores the results in the data structure JDIGI [HITS].

- ***GUOUT*** outputs the relevant data structures for the current event [IOPA].

Other user routines called during the tracking phase triggered by GTREVE should be mentioned for completeness:

- The hadronic processes activated by default for the tracking of hadrons in GEANT3 have originally been extracted from the program TATINA. In the subroutines ***GUPHAD*** and ***GUHADR*** [TRAK] the user may select, instead of TATINA, the hadronic part of the program GHEISHA [PHYS].

- After each tracking step along a given track in a given medium, control is given to the subroutine ***GUSTEP***. From the information available in labelled common blocks the user is able to take the appropriate action, such as storing a hit or transfering a secondary product either in the stack JSTAK or in the event structure JVERTX/JKINE.

- The subroutine ***GUSWIM*** is called by various tracking routines to select the appropriate code for transport of the particle over the given tracking step.

- The magnetic field, unless constant with no X- or Y-component, has to be described in the subroutine ***GUFLD***.

5. Termination

The termination phase is under the control of the user [BASE 300]. For trivial applications it may simply consist of a call to the subroutine GLAST which computes and prints some statistical information (time per event, use of dynamic memory, etc.).

Author(s) : F.Bruyant
Origin : GEANT3

Submitted: 01.10.84
Revised: 20.05.86

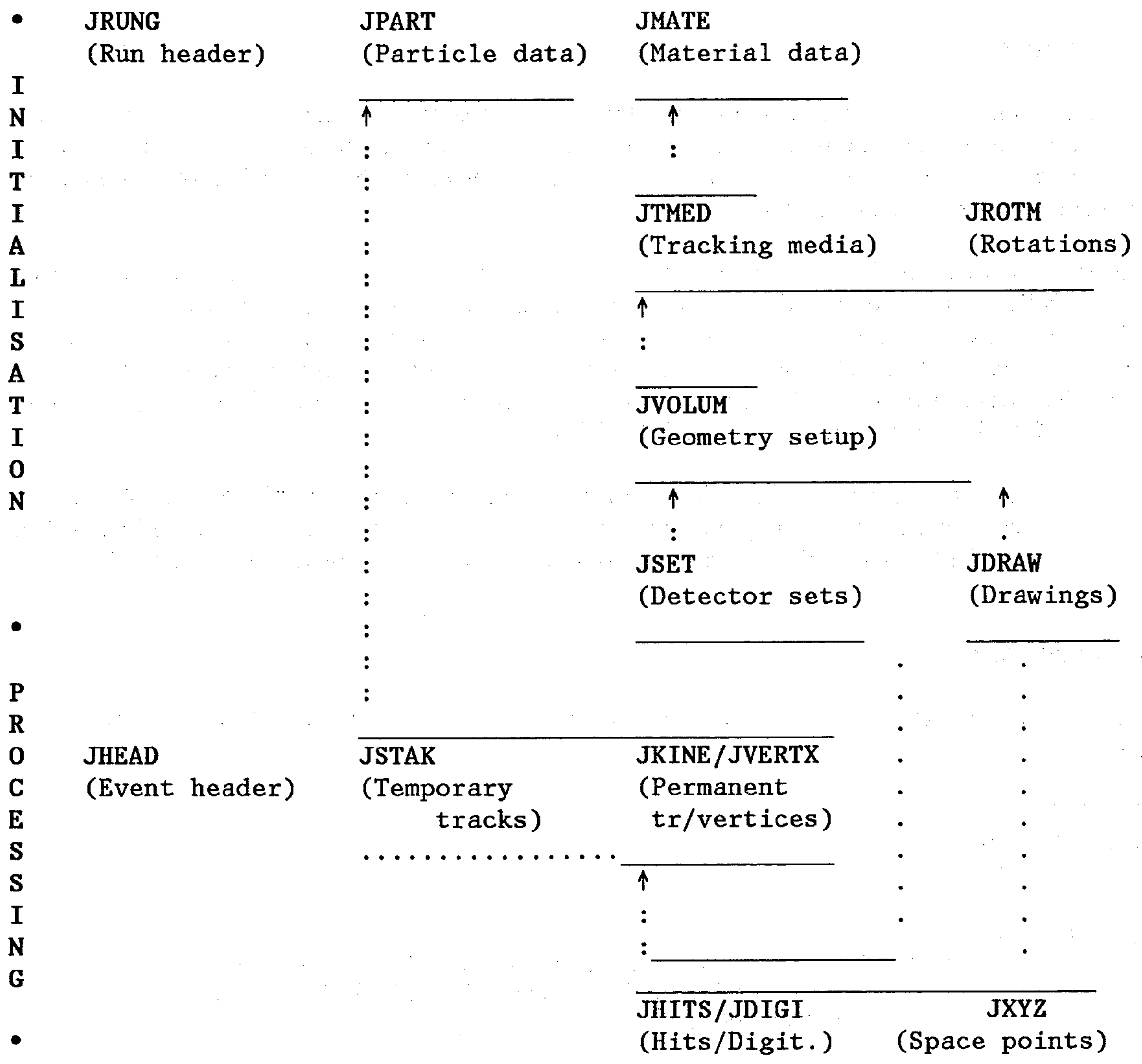
Simplified Program Flow Chart

```
MAIN (user)
 |
 |— GZEBRA  initialisation of ZEBRA system, dynamic core allocation
 |— UGINIT   (user)
 |   |
 |   |— GINIT    initialisation of GEANT3 variables
 |   |— GFFGO    interpretation of data cards
 |   |— GZINIT   initialisation of ZEBRA core divisions and link areas
 |   |— GPART    creation of the 'particle' data structure JPART
 |   |— GMATE    creation of the 'material' data structure JMATE
 |   |— 'user code'  description of the geometrical setup, of the sensitive detectors
 |   |                 creation of data structures JVOLUM, JTMED, JROTM, JSETS
 |   |— GPHYSI preparation of cross-section and energy loss tables for all used materials
 |
 |— GRUN  loop over events
 |  ┌┬┐
 |  ||
 |  ||— GTRIGI   initialisation for event processing
 |  ||— GTRIG    event processing
 |  ||   |
 |  ||   |— GUKINE (user)  generation (or input) of event initial kinematics
 |  ||   |— GUTREV (user)
 |  ||   |     |— GTREVE (loop over tracks, including any secondaries generated)
 |  ||   |          |— GUTRAK (user)
 |  ||   |               |— GTRACK  control tracking of current track
 |  ||   |                    |— GMEDIA  find current volume/tracking medium
 |  ||   |                    |— GTVOL   loop over successive media seen by the particle
 |  ||   |                         |— GTGAMA/GTELEC/.... tracking of particle according to type
 |  ||   |                              |—GUSTEP (user) recording of hits in data structure JHITS
 |  ||   |                                                and of space points in data structure JXYZ
 |  ||   |— GUDIGI  computation of digitisations and recording in data structure JDIGI
 |  ||   |— GUOUT    output of current event
 |  ||
 |  └┴— GTRIGC  clearing of memory for next event
 |
 |— UGLAST  (user)
 |   |
 |   |—  GLAST    standard GEANT3 termination.
 |
stop
```

Author(s) : F.Bruyant, M.Maire
Origin : GEANT3

Submitted: 01.10.84
Revised: 20.08.87

The Data Structures and their Relationship



Detail of structures:

JRUNG	[BASE 299]				
JPART	[CONS 399]	JMATE	[CONS 199]		
JROTM	[GEOM 299]	JTMED	[CONS 299]		
JVOLUM	[GEOM 199]				
JSET	[HITS 199]	JDRAW	[DRAW 399]		
JHEAD	[BASE 299]	JKINE	[KINE 199]	JVERTX	[KINE 199]
JSTAK	[TRAK 399]				
JDIGI	[HITS 399]	JHITS	[HITS 299]	JXYZ	[TRAK 499]

Author(s) :
Origin : GEANT2/3

Submitted: 01.10.84
Revised: 17.08.87

Overview of COMMON Blocks

1. Introduction

The communication between program segments of the GEANT3 system is ensured by the contents of the data structures and by the definition of 'long range' variables in several common blocks. In addition, within the program segments, the subroutines communicate with each other through explicit arguments and through the common block variables.

The data structures are described in separate papers. Here, the main features of the common blocks used in GEANT3 are summarized, with special mention of the variables initialized in GINIT and of the possibility to override them through data cards [BASE040]. In most of the cases there is a correspondance between a given data structure and a given common block, where the current contents of the banks is dumped. The labelled common blocks are accessible through Patchy sequences identified by the name of the COMMON. They are defined in the Patch GCDES.

Note Unless otherwise specified the long range variables are initialized in GINIT. When non zero, default values are quoted between brackets. If the value may be modified through a standard data card the card keyword is also given in **bold characters** between brackets.

2. Dynamic core

The GEANT3 data structures are stored in the common /GCBANK/ accessible through the following Patchy sequence.

+SEQ,GCBANK

```
 +KEEP,GCLINK
       COMMON/GCLINK/JDIGI,JDRAW,JHEAD,JHITS, JKINE, JMATE, JPART,
      +              JROTM,JRUNG,JSET, JSTAK, JGSTAT,JTMED, JTRACK,
      +              JVERTX,JVOLUM,JXYZ
C
 +KEEP,GCBANK
       COMMON/GCBANK/NZEBRA,GVERSN,ZVERSN,IXSTOR,IXDIV,IXCONS,FENDQ(16),
      +              LMAIN, LR1,   WS(9000)
       DIMENSION IQ(1),Q(1),LQ(8000)
       EQUIVALENCE (Q(1),IQ(1),LQ(9)),(LQ(1),LMAIN)
 +SEQ,GCLINK
C
```

The /GCLINK/ variables are pointers to the GEANT3 data structures. They belong to a permanent area declared in GZINIT. The pointer JTRACK is reserved for future applications.

3. Other labelled COMMON blocks

+SEQ,GCCUTS Cuts for tracking

```
      COMMON/GCCUTS/CUTGAM,CUTELE,CUTNEU,CUTHAD,CUTMUO,BCUTE,BCUTM,
     +              DCUTE,DCUTM,PPCUTM,TOFMAX,GCUTS(5)
```

CUTGAM	Kinetic energy cut for gammas	(0.001, ***CUTS***)
CUTELE	Kinetic energy cut for electrons	(0.001, ***CUTS***)
CUTNEU	Kinetic energy cut for neutral hadrons	(0.01, ***CUTS***)
CUTHAD	Kinetic energy cut for charged hadrons	(0.01, ***CUTS***)
CUTMUO	Kinetic energy cut for muons	(0.01, ***CUTS***)
BCUTE	Kinetic energy cut for electron Bremsstrahlung	(CUTGAM, ***CUTS***)
BCUTM	Kinetic energy cut for muon Bremsstrahlung	(CUTGAM, ***CUTS***)
DCUTE	Kinetic energy cut for electron delta-rays	(10000., ***CUTS***)
DCUTM	Kinetic energy cut for muon or hadron delta-rays	(10000., ***CUTS***)
PPCUTM	Total energy cut for e^+ e^- pair production by muons	(0.01, ***CUTS***)
TOFMAX	Tracking cut on time of flight integrated from primary interaction time	(1.E+10, ***CUTS***)
GCUTS	For user applications	(***CUTS***)

Note The cuts BCUTE, BCUTM and DCUTE, DCUTM are given the respective default values CUTGAM and CUTELE. Only experienced users should make use of the facility offered by the data card CUTS to change BCUTE, DCUTE, BCUTM and DCUTM.

+SEQ,GCDRAW Variables used by the drawing package

```
      COMMON/GCDRAW/NUMNOD,MAXNOD,NUMND1,LEVVER,LEVHOR,MAXV,IPICK,
     +              MLEVV,MLEVH,NWCUT,JNAM,JMOT,JXON,JBRO,JDUP,JSCA,JDVM,JPSM,
     +              JNAM1,JMOT1,JXON1,JBRO1,JDUP1,JSCA1,JULEV,JVLEV,LOOKTB(16),
     +              GRMAT0(10),GTRAN0(3),IDRNUM,GSIN(41),GCOS(41),SINPSI,COSPSI,
     +              GTHETA,GPHI,GPSI,GU0,GV0,GSCU,GSCV,NGVIEW,
     +              ICUTFL,ICUT,CTHETA,CPHI,DCUT,NSURF,ISURF,
     +              GZUA,GZVA,GZUB,GZVB,GZUC,GZVC,PLTRNX,PLTRNY,
     +              LINATT,LINATP,ITXATT,ITHRZ,IPRJ,DPERS,ITR3D,IPKHIT,IOBJ,
     +              LINBUF,MAXGU,MORGU,MAXGS,MORGS,MAXTU,MORTU,MAXTS,MORTS,
     +              IGU,IGS,ITU,ITS,NKVIEW,IGVIEW,
     +              NOPEN,IGMR,IPIONS,ITRKOP,
     +              DDUMMY(19)
```

NUMNOD number of nodes in non-optimized tree

MAXNOD max number of nodes of non-optimized tree (init : MIN(NLEFT/16,2000))

NUMND1 number of nodes in optimized tree

LEVVER vertical level in the tree being scanned by tree routines

LEVHOR horizontal level in the tree being scanned by tree routines

MAXV max vertical levels in the tree to be scanned by tree routines

IPICK node picked by GDTREE

MLEVV number of vertical levels in the last tree scanned

MLEVH number of horizontal levels in the last tree scanned

NWCUT max workspace allocated by cut routines (init : 5000)

JNAM pointer used by the tree routines

	. . . .
JVLEV	pointer used by the tree routines
LOOKTB	colour look-up table (init : LOOKTB(I) = I , I=1,16)
GRMAT0	rotation matrix saved by GDRVOL (init : unitary matrix)
GTRAN0	translation matrix saved by GDRVOL (init : 0., 0., 0.)
IDRNUM	flag for GDRAW, set to 1 when called by GDRVOL (init : 0)
GSIN	sine table (40 divisions of a full circle)
GCOS	cosine table (40 divisions of a full circle)
SINPSI	sin(gpsi*degrad)
COSPSI	cos(gpsi*degrad)
GTHETA	theta angle (init : 45.)
GPHI	phi angle (init : 135.)
GPSI	psi angle (init : 0.)
GU0	u-coordinate of volum origin (init : 10.)
GV0	v-coordinate of volum origin (init : 10.)
GSCU	scale factor for u-coordinate (init : 0.015)
GSCV	scale factor for v-coordinate (init : 0.015)
NGVIEW	flag informing GDFR3D and GD3D3D if the view point has changed (init : 0)
ICUTFL	flag informing GDRAW if it was called by 'cut' routines
ICUT	axis along which the cut is performed (1,2,3 or 0 if no cut)
CTHETA	theta angle of cut supplied to GDRAWX (used by GDCUT)
CPHI	phi angle of cut supplied to GDRAWX (used by GDCUT)
DCUT	coordinate value (along axis ICUT) at which the cut is performed
NSURF	number of surfaces stored in SURF
ISURF	pointer for array SURF
GZUA	zoom parameter (horizontal scale factor) (init : 1.)
GZVA	zoom parameter (vertical scale factor) (init : 1.)
GZUB	zoom parameter (init : 0.)
GZVB	zoom parameter (init : 0.)
GZUC	zoom parameter (init : 0.)
GZVC	zoom parameter (init : 0.)
PLTRNX	screen and plotter X range, PLTRNX x PLTRNY cm. (init : 20.)
PLTRNY	screen and plotter Y range, PLTRNX x PLTRNY cm. (init : 20.)
LINATT	current line attributes (init : colour = 1, width = 1, style = 1, fill = 0)
LINATP	permanent line attributes (init : LINATT)
ITXATT	current text attributes (init : colour = 1, width = 1)
ITHRZ	string containing the status of THRZ option of GDOPT (init : 'OFF ')
IPRJ	string containing the status of PROJ option of GDOPT (init : 'PARA')
DPERS	distance of the view point from the origin (used with perspective) (init : 1000.)
ITR3D	track being scanned (used together with THRZ option)
IPKHIT	flag for GPHITS (normally 0), if >0 then print only hit nr. IPKHIT
IOBJ	type of the object being drawn (detector, track, hit, etc.) (init : 0)
LINBUF	flag informing GDRAWV if line buffering is wanted or not (init : 0)
MAXGU	current physical number of words for graphic unit banks
MORGU	number of words to be pushed in graphic unit banks
MAXGS	current physical number of words for graphic segment banks
MORGS	number of words to be pushed in graphic segment banks
MAXTU	current physical number of words for text unit banks
MORTU	number of words to be pushed in text unit banks
MAXTS	current physical number of words for text segment banks
MORTS	number of words to be pushed in text segment banks
IGU	pointer to current graphic unit bank
IGS	pointer to current graphic segment bank
ITU	pointer to current text unit bank
ITS	pointer to current text segment bank
NKVIEW	number of view data banks (init : 0)

IGVIEW	current view number or 0 for screen (init : 0)
NOPEN	number of logical objects opened by PIONS
IGMR	flag informing if APOLLO – GMR is being used (init : 0)
IPIONS	flag informing if PIONS is being used (init : 0)
ITRKOP	string containing the status of TRAK option of GDOPT (init : 'LINE')
DDUMMY	array of dummy words

+SEQ,GCFLAG Flags and variables to control the run

```
      COMMON/GCFLAG/IDEBUG,IDEMIN,IDEMAX,ITEST, IDRUN, IDEVT, IEORUN,
     +              IEOTRI,IEVENT,ISWIT(10),IFINIT(20),NEVENT,NRNDM(2)
```

IDEBUG	Flag set internally to 1 to activate debug if IEVENT (below)	
IDEMIN	is greater or equal to IDEMIN	***(DEBU)***
IDEMAX	and less or equal to IDEMAX	***(DEBU)***
ITEST	Flag to request printing of IEVENT, IDEVT and NRNDM (below) every ITEST events	***(DEBU)***
IDRUN	Current user run number	(1, ***RUNG***)
IDEVT	Current user event number	***(RUNG)***
IEORUN	Flag to terminate run if non zero	
IEOTRI	Flag to abort current event if non zero	
IEVENT	Current event sequence number	(1)
ISWIT	Flags reserved for user in relation to debug	***(SWIT)***
IFINIT	Flags used by system for initialisation	
NEVENT	Number of events to be processed	(10000000, ***TRIG***)
NRNDM	Initial random number value in two bytes of NBIT/2 bits (NBIT as defined in /GCNUM/ below) NRNDM(1) left byte, NRNDM(2) right byte If NRNDM(2) is 0, the default generator seed is used.	***(RNDM)***

+SEQ,GCJLOC JMATE substructure pointers for current material

```
      COMMON /GCJLOC/ NJLOC(2),JTM, JMA, JLOSS, JPROB, JMIXT, JPHOT, JANNI,
     +                JCOMP, JBREM, JPAIR, JDRAY, JPFIS, JMUNU, JHADR
```

Self-explanatory [CONS 199]

+SEQ,GCKINE Kinematics of current track

```
      COMMON /GCKINE/ IKINE, PKINE(10), ITRA, ISTAK, IVERT, IPART, ITRTYP,
     +                NAPART(5),AMASS,CHARGE,TLIFE,VERT(3),PVERT(4),IPAOLD
```

IKINE		(0, ***KINE***)
PKINE	for user applications	(1.E+10, ***KINE***)
ITRA	Current track number	
ISTAK	Current stack-track number	
IVERT	Current vertex number	

IPART	Current particle number
ITRTYP	'Tracking type' of current particle
NAPART	Name of current particle
AMASS	Mass of current particle
CHARGE	Charge of current particle
TLIFE	Life-time of current particle
VERT	Coordinates of origin vertex for current track
PVERT	Track kinematics at origin vertex
IPAOLD	Particle number of the last track.

+SEQ,GCKING Kinematics of generated secondaries

```
      COMMON /GCKING/ KCASE, NGKINE, GKIN(5,100), TOFD(100)
```

KCASE	Mechanism having generated the secondary particles
NGKINE	Number of generated secondaries
GKIN(1,I)	Px of I-th secondary
GKIN(2,I)	Py " "
GKIN(3,I)	Pz " "
GKIN(4,I)	E " "
GKIN(5,I)	Particle type " "
TOFD(I)	Time delay introduced by the interaction.

+SEQ,GCLINK See GCBANK above

+SEQ,GCLIST Various system and user lists

```
     COMMON/GCLIST/NHSTA, NGET, NSAVE, NSETS,
    +              NPRIN, NGEOM, NVIEW, NPLOT, NSTAT,
    +              LHSTA(20), LGET(20), LSAVE(20), LSETS(20),
    +              LPRIN(20), LGEOM(20), LVIEW(20), LPLOT(20), LSTAT(20)
```

NHSTA	Number of histograms declared on data card ***HSTA***
NGET	Number of data structures declared on data card ***GET***
NSAVE	Number of data structures declared on data card ***SAVE***
NSETS	Number of items described on data card ***SETS***
NPRIN	Number of items described on data card ***PRIN***
NGEOM	Number of items described on data card ***GEOM***
NVIEW	Number of items described on data card ***VIEW***
NPLOT	Number of items described on data card ***PLOT***
NSTAT	Number of items described on data card ***STAT***
LHSTA,...,LSTAT	Corresponding user lists of items (HSTA,...,STAT)

LSTAT(1) is reserved by the system for volume statistics.

+SEQ,GCMATE

Parameters of current material

```
COMMON /GCMATE/ NMAT, NAMATE(5), A, Z, DENS, RADL, ABSL
```

NMAT	Current material number
NAMATE	Name of current material
A	Atomic weight of current material
Z	Atomic number of current material
DENS	Density of current material
RADL	Radiation length of current material
ABSL	Absorption length of current material

+SEQ,GCMULO

– See CONS-199

Precomputed quantities for multiple scattering and energy binning for JMATE banks.

```
 COMMON/GCMULO/SINMUL(101),COSMUL(101),SQRMUL(101),OMCMOL,CHCMOL
+               ,ELOW(90),DEINV(9)
```

+SEQ,GCMZFO

I/O descriptors of GEANT banks (self explanatory)

```
 COMMON/GCMZFO/  IOMATE,IOPART,IOTMED,IOSEJD,IOSJDD,IOSJDH,
+                IOSTAK,IOMZFO(13)
```

+SEQ,GCNUM

Current number for various items

```
 COMMON/GCNUM/  NMATE, NVOLUM, NROTM, NTMED, NTMULT,
+               NTRACK,NPART, NSTMAX,NVERTX,NHEAD, NBIT
```

NMATE	Number of Materials
NVOLUM	Number of Volumes
NROTM	Number of Rotation matrices
NTMED	Number of Tracking media
NTMULT	Number of tracks processed in current event (including secondaries), reset to 0 for each event
NTRACK	Number of Tracks in JKINE mother bank for current event
NPART	Number of Particle banks
NSTMAX	Maximum number of tracks in stack JSTAK for current event, reset to 0 for each event
NVERTX	Number of Vertices in JVERTX mother bank for current event
NHEAD	Number of data words in the JHEAD bank (10)
NBIT	Number of bits per word (initialized in GINIT through ZEBRA)

+SEQ,GCONST Natural constants (or considered as such!)

```
COMMON/GCONST/PI,  TWOPI, PIBY2, DEGRAD,RADDEG,CLIGHT,BIG, EMASS
```

PI Number PI (ACOS(−1.))
TWOPI (2.*PI)
PIBY2 (PI/2.)
DEGRAD Degree to radian conversion factor (PI/180.)
RADDEG Radian to degree conversion factor (180./PI)
CLIGHT Light velocity (2.99792458E+10)
BIG Arbitrary large number (1.E+10)
EMASS Electron mass (0.5110034E−03)

+SEQ,GCPHYS – See PHYS 010

Control of physics processes.

```
COMMON/GCPHYS/IPAIR, SPAIR, SLPAIR,ZINTPA,STEPPA
+            ,ICOMP, SCOMP, SLCOMP,ZINTCO,STEPCO
+            ,IPHOT, SPHOT, SLPHOT,ZINTPH,STEPPH
+            ,IPFIS, SPFIS, SLPFIS,ZINTPF,STEPPF
+            ,IDRAY, SDRAY, SLDRAY,ZINTDR,STEPDR
+            ,IANNI, SANNI, SLANNI,ZINTAN,STEPAN
+            ,IBREM, SBREM, SLBREM,ZINTBR,STEPBR
+            ,IHADR, SHADR, SLHADR,ZINTHA,STEPHA
+            ,IMUNU, SMUNU, SLMUNU,ZINTMU,STEPMU
+            ,IDCAY, SDCAY, SLIFE ,SUMLIF,DPHYS1
+            ,ILOSS, SLOSS, SOLOSS,STLOSS,DPHYS2
+            ,IMULS, SMULS, SOMULS,STMULS,DPHYS3
```

IPAIR 1 = pair production with effective generation of secondaries
2 = same without generation of secondaries (2, ***PAIR***)
ICOMP 1 = Compton scattering with effective generation of electron
2 = same without generation of electron (2, ***COMP***)
IPHOT 1 = photo-electric effect with effective generation of electron
2 = same without generation of electron (2, ***PHOT***)
IPFIS 0 = No photofission (0, ***PFIS***)
1 = photofission with effective generation of secondaries
2 = same without generation of secondaries
IDRAY 0 = no delta ray production (0, ***DRAY***)
1 = delta ray production with effective generation of electron
2 = same without generation of electron
IANNI 1 = positron annihilation with effective generation of secondaries
2 = same without generation of secondaries (2, ***ANNI***)
IBREM 1 = Bremsstrahlung with effective generation of gamma
2 = same without generation of gamma (2, ***BREM***)
IHADR 1 = hadron interactions with effective generation of secondaries
2 = same without generation of secondaries (2, ***HADR***)
IMUNU 0 = no muon nuclear interaction (0, ***MUNU***)
1 = muon nuclear interaction with effective generation of secondaries
2 = same without generation of secondaries

IDCAY	Flag to inhibit decay if zero	(1)
ILOSS	0 = no energy loss	
	1 = average energy loss	
	2 = Landau fluctuations	(2, ***LOSS***)
IMULS	0 = no multiple scattering	
	1 = Gaussian scattering	(1, ***MULS***)
	2 = Moli'ere scattering	

For more details on IDRAY and ILOSS see [BASE 040].

For all other variables see [PHYS 010].

+SEQ,GCPOLY Internal flags for polygon and polycone shapes

IZSEC	Z-section number
IPSEC	Phi-sector number

+SEQ,GCPUSH Initial and incremental size of some mother banks

```
COMMON/GCPUSH/NCVERT,NCKINE,NCJXYZ,NPVERT,NPKINE,NPJXYZ
```

NCVERT	Initial size of mother bank JVERTX (5)
NCKINE	Initial size of mother bank JKINE (50)
NCJXYZ	Initial size of mother bank JXYZ (50)
NPVERT	Increment for size of mother bank JVERTX (5)
NPKINE	Increment for size of mother bank JKINE (10)
NPJXYZ	Increment for size of mother bank JXYZ (10)

+SEQ,GCRZ

```
COMMON /GCRZ1/ NRECRZ,NRGET ,NRSAVE,LRGET(20),LRSAVE(20)
COMMON /GCRZ2/ RZTAGS
```

NRECRZ	Record size (argument of RZMAKE)
NRGET	Number of data structures declared on data card RGET
NRSAVE	Number of data structures declared on data card RSAV
LRGET,LRSAVE	Corresponding user lists of items
RZTAGS	Key names (argument of RZMAKE)

+SEQ,GCSETS Identication of current sensitive detector

```
COMMON /GCSETS/ IUSET, IUDET, ISET, IDET, IDTYPE, NVNAME, NUMBV(10)
```

IUSET	Set identifier
IUDET	Detector identifier

ISET	Position of set in bank JSET
IDET	Position of detector in bank JS = LQ(JSET − IDET)
IDTYPE	User defined detector type
NVNAME	Number of elements in NUMBV
NUMBV	List of volume numbers to identify the detector

+SEQ,GCTATI Temporary storage of secondaries, for TATINA only

```
    COMMON/GCTATI/NSEC, KIND(30),EN(30),PL(30),PT(30),THETA(30),PHI(30)
```

NSEC	Number of generated secondaries
KIND	'Tracking type' of secondaries
EN	Energy of secondaries
PL	Longitudinal momentum of secondaries
PT	Transverse momentum of secondaries
THETA	Angles of secondaries
PHI	Angles of secondaries

+SEQ,GCTIME Execution time control

```
   COMMON/GCTIME/TIMINT, TIMEND, ITIME, IGDATE, IGTIME
```

TIMINT	Total time left after initialization	(System, *TIME*)
TIMEND	Time required for program termination phase	(1., *TIME*)
ITIME	Test on time left done every ITIME events	(1, *TIME*)
IGDATE	Date of the day YYMMDD integer	(e.g. 860124)
IGTIME	Time of the day HHMM integer	(e.g. 1425)

+SEQ,GCTMED Parameters of current tracking medium

```
      COMMON /GCTMED/ NUMED, NATMED(5), ISVOL, IFIELD, FIELDM,
   +               TMAXFD, DMAXMS, DEEMAX, EPSIL, STMIN,
   +               CFIELD, CMULS, IUPD, ISTPAR, NUMOLD
```

NUMED	Current tracking medium number
NATMED	Name of current tracking medium
ISVOL	Sensitive volume flag (if non zero)
IFIELD	Field map type (0 if no field)
FIELDM	Maximum field
TMAXFD	Maximum field turning angle in one step
DMAXMS	Maximum multiple scattering displacement in one step
DEEMAX	Maximum energy loss gradient in one step
EPSIL	Boundary crossing accuracy
STMIN	Minimum step size by energy loss or by multiple scattering
CFIELD	Constant for field step evaluation
CMULS	Constant for multiple scattering step evaluation
IUPD	0 If medium change, (1 otherwise)

ISTPAR 0 If standard tracking parameters
NUMOLD Numed of the last medium.

+SEQ,GCTRAK Track parameters at current point

```
      COMMON /GCTRAK/ VECT(7), GETOT, GEKIN, VOUT(7), NMEC,
     +                LMEC(30), NAMEC(30), NSTEP, MAXNST, DESTEP,
     +                SAFETY, SLENG, STEP, SNEXT, SFIELD, SNPHYS,
     +                TOFG, GEKRAT, IGNEXT, INWVOL, ISTOP, IDECAD, IEKBIN
```

VECT Current track parameters (X,Y,Z,P_x/P,P_y/P,P_z/P,P)
GETOT Current track total energy
GEKIN Current track kinetic energy
VOUT Same as VECT after extrapolation
NMEC Number of mechanisms for current step
LMEC List of mechanism indices for current step
NAMEC Mechanism names - See below
NSTEP Number of steps so far
MAXNST Maximum number of steps allowed (default = 10000)
DESTEP Total energy lost in current step
SAFETY Overestimated distance to closest medium boundary
SLENG Track length at current point
STEP Size of curent tracking step
SNEXT Straight distance to next current medium boundary
SFIELD Field turning angle step size evaluation
SNPHYS Remaining distance to next physics process
(for GTGAMA and GTNEUT only)
TOFG Current time of flight
GEKRAT Interpolation factor in table ELOW
IGNEXT Flag set to 1 when SNEXT has to be recomputed
INWVOL Flag set to 1 when entering a new volume, or when starting a new track
to 2 when leaving a volume and
to 3 when leaving the experimental setup.
(Otherwise 0)
ISTOP Flag set to 1 when track looses its identity
to 2 when energy below cut
IDECAD Energy decade [CONS 199]
IEKBIN Current kinetic energy bin in table ELOW

List of mechanisms considered at tracking time:

```
      DATA MEC/'NEXT','MULS','LOSS','FIEL','DCAY','PAIR','COMP','PHOT'
     +        ,'BREM','DRAY','ANNI','HADR','ECOH','EVAP','FISS','ABSO'
     +        ,'ANNH','CAPT','EINC','INHE','MUNU','TOFM','PFIS','NONE'
     +        ,'NONE','NONE','NONE','NONE','NONE','STOP'/
```

+SEQ,GCTRA2 Later in GCTRAK

```
      COMMON /GCTRA2/ DESTEL, UPWGHT
```

DESTEL Energy lost in current step by continuous energy loss
UPWGHT User Particle weight

+*SEQ,GCUNIT* Description of logical units

```
    COMMON/GCUNIT/LIN, LOUT, NUNITS, LUNITS(5)
    COMMON/GCMAIL/CHMAIL
    CHARACTER*132 CHMAIL
```

LIN Input unit to read data cards
LOUT Line printer output unit
NUNITS Number of additional units
LUNITS List of additional units
CHMAIL Character string containing the message to be printed by GMAIL

LIN and LOUT are defined in GINIT through ZEBRA

NUNITS and LUNITS are reserved for user declared FZ-type ZEBRA files.

+*SEQ,GCVOLU* Multi-level current volume description

```
     COMMON/GCVOLU/NLEVEL,NAMES(15),NUMBER(15),
    +             LVOLUM(15),LINDEX(15),GTRAN(3,15),GRMAT(10,15),
    +             NGPAR(15),GPAR(50,15),GONLY(15),GLX(3),
    +             NJNEXT(2),JNEXT(15),INEXT(15),NNEXT(15)
```

NLEVEL Level number at which the last medium search stopped.
NAMES Volume names at each level.
NUMBER User volume numbers at each level.
LVOLUM System volume numbers at each level.
LINDEX Physical tree volume indices at each level.
GTRAN X,Y,Z offsets of the cumulative coordinate transformation from the master system to the system at each level.
GRMAT Rotation matrix elements for the cumulative transformation from the master system to the system at each level. GRMAT(10,LEVEL) equal to 0.0 indicates the null rotation.
NGPAR Number of parameters for the shape at each level.
GPAR Shape parameters.
GONLY Uniqueness flags at each level.
GLX Current point in local coordinates system (local use only!)
(XX)NEXT.. For control of search sequence.

+*SEQ,GCVOL2* Back-up for /GCVOLU/

+*SEQ,GSECTI* Renato Bruno laboris

Author(s) : F. Bruyant, M. Maire
Origin : GEANT2/3

Submitted: 01.10.84
Revised: 18.08.87

Summary of Data Cards

1. Introduction

GEANT3 uses the standard FFREAD[5] package to read 'free format' data cards in the routine GFFGO.

The cards currently interpreted by GFFGO can be classified into five categories:

- General control of the run.
- Control of the physics processes.
- Debug and I/O operations.
- User applications.
- Lund event generation.

The data cards are listed below by category with the following information:

- KEY, card keyword, any number of characters truncated to the first 4
- N, maximum expected number of variables NVAR,
- T, TYPE of these variables (I=INTEGER, R=REAL or M=MIXED) and for each variable in turn:
  - VAR.., FORTRAN name
  - Short description (more detail in BASE 030)
  - Labelled COMMON where it is stored, and
  - Default value usually from GINIT.

When a card is decoded, the values entered by the user without explicit assignment are assigned to the variables in order. The number of values can be less than NVAR. In case of a MIXED type the values entered have to be in agreement with the default of the corresponding FORTRAN variable names.

Example of data card: RUNG 5 201
to preset the current run and event number to 5 and 201 respectively.

In batch jobs there is no need for any special termination card and none of the cards mentioned below is mandatory.

2. User defined data cards

Before calling GFFGO the user may define private data cards through calls to FFKEY as follows:

```
CALL FFKEY('KEY',VAR(1),NVAR,'TYPE')
```

They will be interpreted by GFFGO in the same way as the standard cards.

[5] FFREAD User Guide and Reference Manual, CERN DD/US/71, DD/EE/78-2.

3. Summary of GEANT3 Data Cards

KEY	N	T	VAR..	Short description	COMMON	GINIT
	General control of the run:					
HSTA	20	M	LHSTA [BASE 110]	Names of required standard histograms	GCLIST	' '
RNDM	2	I	NRNDM	Initial random number seed (2 words)	GCFLAG	0
RUNG	2	I	IDRUN	User run number	GCFLAG	1
			IDEVT	User event number	GCFLAG	1
TIME	3	M	TIMINT	Time left after initialisation	GCTIME	System
			TIMEND	Time required for termination	GCTIME	1.
			ITIME	Test every ITIME events	GCTIME	1
TRIG	1	I	NEVENT	Number of events to process	GCFLAG	10000000
	Control of physics processes [PHYS 001]:					
ANNI	1	I	IANNI	Annihilation flag	GCPHYS	2
BREM	1	I	IBREM	Bremsstrahlung flag	GCPHYS	2
COMP	1	I	ICOMP	Compton scattering flag	GCPHYS	2
CUTS	16	R		Kinetic energy cuts :		
			CUTGAM	" " for gammas	GCCUTS	0.001
			CUTELE	" " for electrons	GCCUTS	0.001
			CUTNEU	" " for neutral hadrons	GCCUTS	0.01
			CUTHAD	" " for charged hadrons	GCCUTS	0.01
			CUTMUO	" " for muons	GCCUTS	0.01
			BCUTE	" " for electron Bremsstrahlung	GCCUTS	CUTGAM
			BCUTM	" " for muon and hadron Brems.	GCCUTS	CUTGAM
			DCUTE	" " for delta-rays by electrons	GCCUTS	1.E+04
			DCUTM	" " for delta-rays by muons	GCCUTS	1.E+04
			PPCUTM	Total energy cut for pair prod. by muons	GCCUTS	0.01
			TOFMAX	Time of flight cut	GCCUTS	1.E+10
			GCUTS	5 user words	GCCUTS	0.
DRAY	1	I	IDRAY	Delta-ray flag	GCPHYS	0
HADR	1	I	IHADR	Hadronic process flag	GCPHYS	2
LOSS	1	I	ILOSS	Energy loss flag (See Point 4 below)	CGPHYS	2
MULS	1	I	IMULS	Multiple scattering flag	GCPHYS	1
MUNU	1	I	IMUNU	Muon nuclear interaction flag	GCPHYS	0
PAIR	1	I	IPAIR	Pair production flag	GCPHYS	2
PFIS	1	I	IPFIS	Photofission flag	GCPHYS	0
PHOT	1	I	IPHOT	Photo-electric effect flag	GCPHYS	2
	Debug and I/O operations:					
DEBU	3	M	IDEMIN	First event to debug If negative, initialisatio debug	GCFLAG	0
			IDEMAX	Last event to debug	GCFLAG	0
			ITEST	Print control frequency	GCFLAG	0
GET	20	M	LGET	Names of data structures to fetch	GCLIST	' '
PRIN	20	M	LPRIN	User keywords to print data structure	GCLIST	' '
RGET	20	M	LRGET	Names of data structures to fetch (RZ)	GCRZ1	' '
RSAV	10	M	LRSAVE	Names of data structures to save (RZ)	GCRZ1	0
SAVE	20	M	LSAVE	Names of data structures to save	GCLIST	' '
SWIT	10	I	ISWIT	User flags for debug or else	GCFLAG	0

User applications:

KINE	11	M	IKINE	User flag	GCKINE	0
			PKINE	10 user words	GCKINE	1.E+10
SETS	20	M	LSETS	User words for detector sets	GCLIST	' '
STAT	20	M	LSTAT	1 system + 19 user words	GCLIST	' '
PLOT	20	M	LPLOT	User words to control plots	GCLIST	' '
GEOM	20	M	LGEOM	User words to control geom setup	GCLIST	' '
VIEW	20	M	LVIEW	User words to control view banks	GCLIST	' '

Lund event generation:

MSTE	20	I	MSTE	See GLUNDI code	LUDATE	
KTYP	120	I	KTYP	See GLUNDI code	LUDAT2	
PMAS	120	R	PMAS	See GLUNDI code	LUDAT2	
PWID	60	R	PWID	See GLUNDI code	LUDAT2	
IDB	120	I	IDB	See GLUNDI code	LUDAT3	
LUND	2	M	IFLUND	Flavour code (See GLUNDI)	GCLUND	0
		R	ECLUND	Total CMS energy	GCLUND	94.

4. Landau fluctuations versus delta rays

In order to avoid double counting between Landau fluctuations (ILOSS=2) and explicit delta ray generation (IDRAY=0), an automatic protection is put in GEANT3 : If ILOSS = 2 one cannot change the default value for delta ray generation. The different cases are summarized in the table below.

	Landau ON ILOSS = 2 (default)	Landau OFF ILOSS = 1
IDRAY DCUTE DCUTM	0 10 TeV 10 TeV	2 CUTELE CUTELE

Author(s) : R.Brun, F.Bruyant
Origin : GEANT2/3

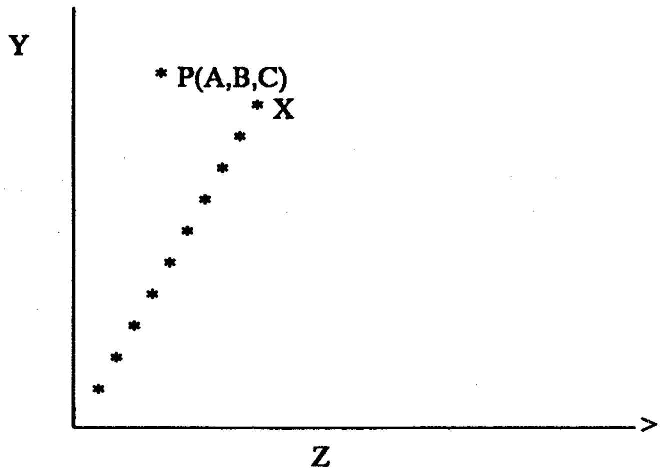
Submitted: 01.10.84
Revised: 13.02.86

The Reference Systems and dimensional Units

1. The MAster Reference System (MARS)

The MAster Reference System (MARS) is determined by the way the user represents the kinematical quantities.

If the axes are labelled (X,Y,Z), then the point P(A,B,C) is represented by



A on axis X
B on axis Y
C on axis Z

The tracking is performed in the MAster Reference System. This implies that the arguments of the user magnetic field routine, space point coordinates and field components, are given in this system.

2. The local reference systems (MRS and DRS)

As explained in GEOM 001, the experimental setup is described by the definition of an 'initial MOTHER' volume inside which 'DAUGHTER' volumes are positioned. Other daughter volumes can be positioned inside these volumes which are promoted as mother volumes and so on, as russian dolls.

This requires the definition of local reference systems, the Mother Reference Systems (MRS, Origin O_m) and the Daughter Reference Systems (DRS, Origin O_d).

The local reference system of the 'initial mother' volume coincides with the MAster Reference System.

The full description of a given detector is usually given in the local reference system of the associated volume.

The transformation of a point from the MRS (V_m) to the DRS (V_d), at any level, requires the knowledge of a rotation matrix R and a translation vector T defined through the relation :

$$(V_d) = [R] (V_m - T)$$

The components of T are the projections of the vector O_mO_d onto the MRS axes.

The rotation matrices are computed from the spherical angles of each of the axes of the daughter reference system (I, II, III) with respect to the mother reference system (1,2,3).

The spherical angles THETA and PHI of a direction D are defined as follows :

THETA	is the angle formed by the axis 3 and D (range : 0 to 180 degrees)
PHI	is the angle formed by the axis 1 and the projection of D onto the plane defined by the axes 1 and 2 (range : 0 to 360 degrees)

Examples are given in GEOM 200.

The various rotation matrices required for a given setup must be defined by the user during the initialisation stage.

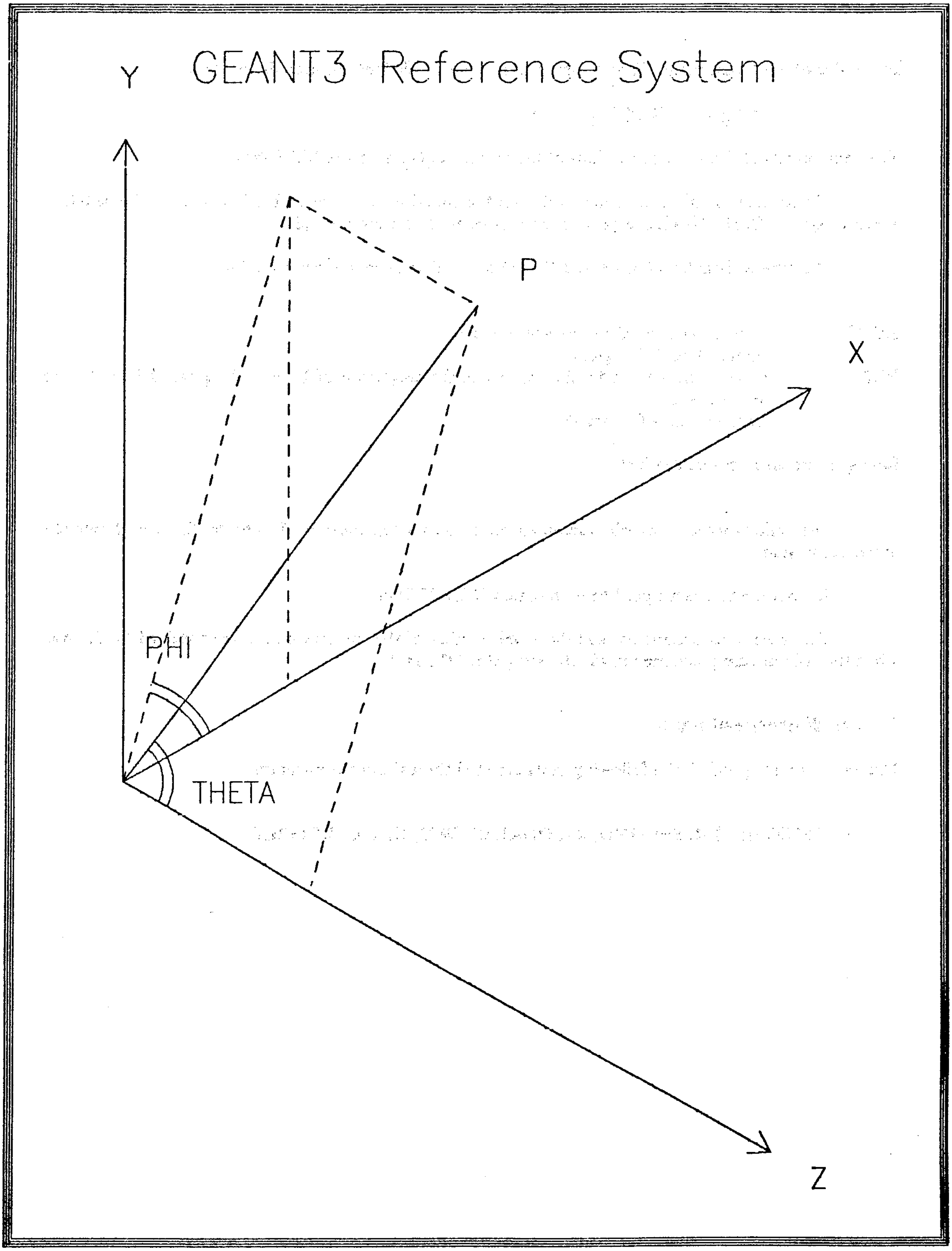
A serial number is assigned to each matrix [GEOM 200].

The translation parameters and the serial number of the rotation matrix are specified by the user when the volumes are positioned inside the setup [GEOM 110].

3. The dimensional units

Unless otherwise specified, the following units are used throughout the program :

- CENTIMETER, SECOND, KILOGAUSS, GEV, GEV/C, DEGREE


GEANT3 Reference System
Y
P
X
PHI
THETA
Z

Author(s) : R. Brun
Origin : GEANT2/3

Submitted: 01.10.84
Revised: 30.05.86

Examples of MAIN Program and User Initialisation

```
      PROGRAM MAIN
C
      PARAMETER (NG=100000,NH=10000)
      COMMON//H(NH)
      COMMON/GCBANK/Q(NG)
C
C        Allocate memory for ZEBRA and HBOOK
      CALL GZEBRA(NG)
      CALL HLIMIT(NH)
C
C        Open GD3 graphic display file (optional)
      CALL TVBGN(3)
C
C        Initialisation phase
      CALL UGINIT
C
C        Processing phase
      CALL GRUN
C
C        Termination phase
      CALL UGLAST
C
      STOP
      END
      SUBROUTINE UGINIT
C
+SEQ,GCLIST
C
C         Initialize GEANT variables
      CALL GINIT
C
C         Read data cards
      CALL GFFGO
C
C         Initialize data structures
      CALL GZINIT
C
C         Initialize drawing package
      CALL GDINIT
C
C         Open I/O buffers
      IF(NGET .GT.0)CALL GOPEN(1,'I',0,IER)
      IF(NSAVE.GT.0)CALL GOPEN(2,'O',0,IER)
C
C         Fetch permanent data structures (if any)
      CALL GGET(1,LGET,-NGET,IDENT,IER)
      IF(IDENT.EQ.0) GO TO 10
C
C         Define standard Particle and Material data
      CALL GPART
      CALL GMATE
C
C         Define the geometrical set-up
      CALL 'user code'
C
C         Compute cross-section and energy loss tables
  10  CALL GPHYSI
C
C         Initialize standard histograms
      CALL GBHSTA
C
      END
```

Author(s) : R.Brun
Origin : GEANT2/3

Submitted: 01.06.83
Revised: 30.05.86

The System Initialisation routines

CALL GZEBRA(NZ) See ZEBRA User Guide

Initializes main store (IXSTOR = 0, NZEBRA = NZ) in common /GCBANK/

NZEBRA = size of common /GCBANK/

The size of the dynamic core LQ is set to NZEBRA −30

CALL GINIT

Presets labelled COMMON block variables to default values.

Preprocessing of various COMMON block variables.

See 'Overview of COMMON blocks' [BASE 030].

CALL GFFGO

Reads a set of data cards with the FFREAD package.

See 'Summary of data cards' [BASE 040]

GFFGO should be called after GINIT.

CALL GZINIT

See ZEBRA User Guide (routine MZSTOR)

Sets index of default division 2 of store IXSTOR to IXDIV = IXSTOR + 2 (Common /GCBANK/)
Allocates the user long term division IXCONS (Common /GCBANK/)
(minimum size 2000, maximum size 8*NZEBRA/10)
The division IXDIV is reserved for the event data structures and the division IXCONS for the initialisation data structures.

Allocates 5200 words of working space.

Initializes the link areas and the Run header bank JRUNG [BASE 299].

Initializes exotic bank formats.

GZINIT should be called after GFFGO.

Image of dynamic store:

LQ(1),LQ(2)/ working space →/ reserve area/ ← div. IXDIV/ system div./ ← div. IXCONS

CALL GDINIT

Initializes the drawing package [DRAW 001]

To be called before the user geometry routine if the user wants to open VIEW banks there.

CALL GPHYSI

Completes the data structure JMATE [PHYS 100]

CALL GBHSTA

Initializes any standard histograms required by the user with the data card HSTA.

The following histogram keywords may be used :

TIME	Time per event
SIZE	Size of division IXDIV per event
MULT	Total number of tracks per event
NTRA	Number of 'long life' tracks per event
STAK	Maximum stack size per event

GBHSTA should be called after GFFGO.

Author(s) : R. Brun
Origin : GEANT2/3

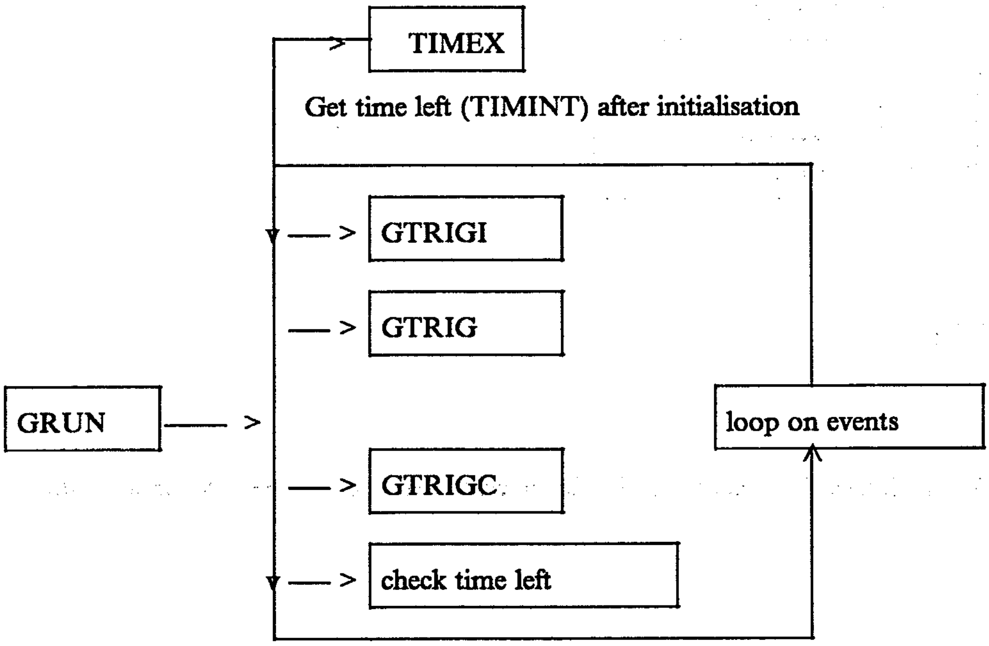
Submitted: 01.06.83
Revised: 23.04.86

Steering routines for Event Processing

The following flow chart is only valid for the 'batch' execution mode. For interactive applications, see section XINT.

CALL GRUN

Main routine to control a run of events



CALL GTRIGI

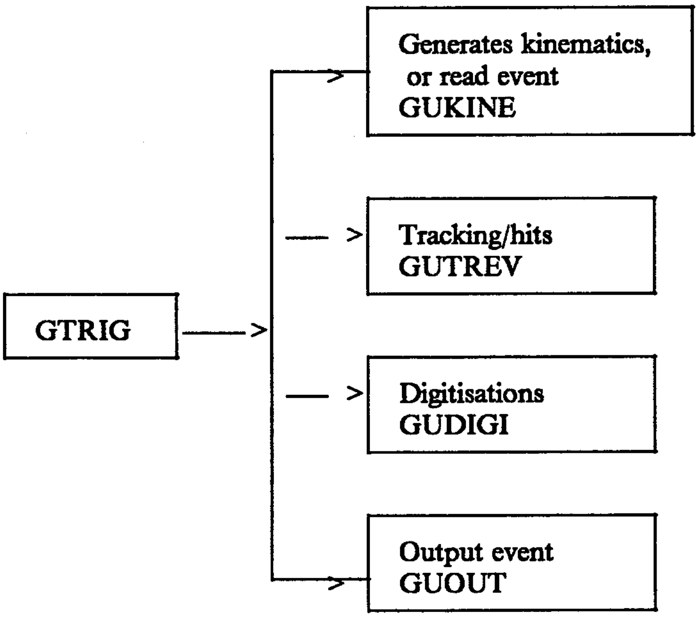
Resets to 0 the flag IEOTRI in /GCFLAG/ and the counters NTRACK and NVERTX in /GCNUM/. Sets the debug flag IDEBUG in /GCFLAG/ to the value required for the current event.

Creates the Header bank for current event [BASE 299].

Prints the sequence number, the event number and the number random generators, under control of the flag ITEST (data card DEBU).

CALL GTRIG

Steering routine to process one event (trigger)



CALL GTRIGC

The event division IXDIV is cleared. The space used by the current event may be used by the next one.

Author(s) : R.Brun, F.Bruyant
Origin : GEANT2/3

Submitted: 01.10.84
Revised: 20.05.86

The banks JRUNG and JHEAD

Run bank JRUNG: 1 user link, 30 data words

```
     LQ(JRUNG-1)   user link

      IQ(JRUNG+1)  IDRUN        Run number
            +2/10)              Reserved for user applications
              +11) creation date for 'INIT' data structures
              +12) creation time for 'INIT' data structures
              +13) creation date for 'KINE'
              +14) creation time for 'KINE'
              +15) creation date for 'HITS'
              +16) creation time for 'HITS'
              +17) creation date for 'DIGI'
              +18) creation time for 'DIGI'

              +21) GEANT version number when 'INIT' created
              +22) ZEBRA version number when 'INIT' created
              +23) GEANT version number when 'KINE' created
              +24) ZEBRA version number when 'KINE' created
              +25) GEANT version number when 'HITS' created
              +26) ZEBRA version number when 'HITS' created
              +27) GEANT version number when 'DIGI' created
              +28) ZEBRA version number when 'DIGI' created
```

Header bank JHEAD: 1 user link, NHEAD(=10) data words

```
      IQ(JHEAD+1)  IDRUN        Run number
               +2) IDEVT        Event number
               +3) NRNDM(1)     Random number bytes at beginning of event
               +4) NRNDM(2)           "      "
            +5/10)              Reserved for user applications
```

Author(s) : R. Brun
Origin : GEANT2/3

Submitted: 01.10.84
Revised: 23.04.86

Example of User Termination and related routines

```
      SUBROUTINE UGLAST
C
+SEQ,GCLIST
C
C         Call standard GEANT termination routine
      CALL GLAST
C
C         Close GD3 graphic display file
      CALL TVEND
C
C         Close I/O buffers
      IF(NGET.NE.0.OR.NSAVE.NE.0) THEN
         CALL GCLOSE(0,IER)
      ENDIF
C
C         Print histograms
      CALL HISTDO
C
      END
```

CALL GLAST

Computes and prints the processing time per event.

Calls MZEND to print the statistics relative to the current run :

If the structure JGSTAT has been initialized, calls GPSTAT [GEOM 700].

Author(s) : R.Brun
Origin : R.Brun, F.Carena

Submitted: 01.10.84
Revised: 28.05.86

Debugging facilities

The flags IDEBUG, ITEST and ISWIT(1 – 10) are available to the user through the common /GCFLAG/ for debug control [BASE 030].

The array ISWIT is user defined and filled through the data card SWIT.

The flag IDEBUG is set to 1 in GTRIGI for the events with sequence number from IDEMIN to IDEMAX, as specified by the user on the data card DEBU.

The flag ITEST, also set by the user on card DEBU, is used by GTRIGI to print the sequence number, the event number and the random numbers at the beginning of an event.

The contents of the data structures can be dumped by calls to the routine GPRINT:

```
CALL GPRINT(NAME,NUMB)
```

NAME	Name of a fan-out Mother Data Structure
NUMB	Link position in Mother of substructure to be printed, 0 for all.

Examples

- CALL GPRINT('KINE',0) prints all banks JKINE
- CALL GPRINT('KINE',8) prints JKINE bank for track 8
- CALL GPRINT('VOLU',0) prints all existing volumes

The following names xxxx are recognized:

DIGI,HITS,KINE,MATE,VOLU,ROTM,SETS,TMED,PART,VERT,JXYZ

GPRINT calls selectively the routines GPxxxx. These routines can also be called directly by the user.

Author(s) : R. Brun
Origin : Same

Submitted: 18.08.87
Revised:

Utility Routines

CALL GLOOK(NAME,IVECT,N,ILOOK)*

Same purpose as the function IUCOMP, for the comparison of a variable NAME (Hollerith, or 4 character word) with a list IVECT of N words.

ILOOK returns the index in IVECT of the first element which compares, 0 if none.

Author(s) : F. Bruyant
Origin : GEANT2/3

Submitted: 01.10.84
Revised: 28.04.86

Introduction to the section CONS

1. The section CONS

The experimental setup is represented by a structure of geometrical volumes. Each volume is given a medium number by the user. Different volumes may have the same medium number [GEOM].

A medium is defined by a set of parameters, the so-called 'tracking medium' parameters, which include reference to the material filling the volume.

The tracking of particles through an experimental setup [TRAK] requires access to the data which describe

the geometrical setup,
the characteristics of the materials used,
the tracking medium parameters, and
the particle properties.

The section CONS contains all routines related to the storage and retrieval of information for the materials, the tracking media and the particles.

2. Materials

The material constants are stored in the data structure JMATE through the routine GMATE which defines the standard table of materials. They can be accessed with the routine GFMATE and printed with the routine GPMATE.

GMATE calls the routine GSMATE for each material in turn. The user may directly use GSMATE instead of, or in addition to, or to partly override, GMATE.

MIXTUREs of basic materials, or COMPOUNDs, molecules with atoms from different basic materials, may also be defined and their characteristics can be stored in the structure JMATE, through calls to the routine GSMIXT. Mixtures of compounds are also accepted.

In addition, some quantities computed during the initialisation of the physics processes are stored in the structure JMATE , such as energy loss and cross-section tables [PHYS]. These can be accessed through the routine GFTMAT and plotted or printed respectively through the routines GPLMAT and GPRMAT.

3. Tracking medium parameters

For each medium in turn, the tracking medium parameters are stored in the data structure JTMED through the routine GSTMED. Details about these parameters are given in [TRAK]. They can be accessed with the routine GFTMED and printed with the routine GPTMED.

The tracking cuts, the physics cuts and the flags which control the physics processes, defined in GINIT and eventually modified through the relevant data cards, are also stored in the structure JTMED. Any of these additional parameters can be modified through the routine GSTPAR.

4. Particles

The particle constants are stored in the data structure JPART through the routine GPART which defines the standard table of particles and, if relevant, the branching ratios and decay modes. The standard particle constants can be accessed with the routine GFPART and printed with the routine GPPART.

GPART calls the routine GSPART (and GSDK for the decays) for each particle in turn. The user may call directly GSPART and GSDK instead of, FILE or in addition to, or to partly override, GPART.

Author(s) : R.Brun, G.N.Patrick
Origin : GEANT2/3

Submitted: 01.06.83
Revised: 18.08.87

Standard Material definition and related routines

CALL GMATE

Stores the following standard material constants in the data structure JMATE.

Material	No.	A	Z	Density	Radiat L	Absorb Len
Hydrogen	1	1.010	1.000	0.071	865.000	790.000
Deuterium	2	2.010	1.000	0.162	757.000	342.000
Helium	3	4.000	2.000	0.125	755.000	478.000
Lithium	4	6.940	3.000	0.534	155.000	120.600
Beryllium	5	9.010	4.000	1.848	35.300	36.700
Carbon	6	12.010	6.000	2.265	18.8	49.900
Nitrogen	7	14.010	7.000	0.808	44.500	99.400
Neon	8	20.180	10.000	1.207	24.000	74.900
Aluminium	9	26.980	13.000	2.700	8.900	37.200
Iron	10	55.850	26.000	7.870	1.760	17.100
Copper	11	63.540	29.000	8.960	1.430	14.800
Tungsten	12	183.850	74.000	19.300	0.350	10.300
Lead	13	207.190	82.000	11.350	0.560	18.500
Uranium	14	238.030	92.000	18.950	0.320	12.000
Air	15	14.610	7.300	1.205E-3	30423	67500.00
Vacuum	16	1.E-16	1.E-16	1.E-16	1.E+16	1.E+16

Note If the user does not need all the above materials, or needs more materials, or wants to override the standard, the routine GSMATE can be called instead to define materials one at a time.

All data, except for nuclear absorption lengths, are taken from M. Aguilar-Benitez. The absorption lengths, kept provisionally for backward compatibility, are no more used.

*) M. Aguilar-Benitez et al, Review of Particle Properties, Rev. Mod. Phys. 56(1984).

```
CALL GSMATE(IMATE,NAMATE,A,Z,DENS,RADL,ABSL,UBUF,NWBUF)
```

Stores the constants for the material IMATE in the data structure JMATE.

IMATE	user material number
NAMATE	material name (up to 20 characters ended by $)
A	atomic weight
Z	atomic number
DENS	density in gr/cm^3
RADL	radiation length
ABSL	absorption length
UBUF	array of NWBUF additional user parameters
NWBUF	

```
CALL GFMATE(IMATE,NAMATE*,A*,Z*,DENS*,RADL*,ABSL*,UBUF*,NWBUF*)
```

Extracts the material constants for the material IMATE from the data structure JMATE.

```
CALL GPMATE(IMATE)
```

Prints the material constants

IMATE	for the material IMATE. (for all materials if IMATE = 0)

Author(s) : R.Brun, M.Maire
Origin : M.Maire

Submitted: 23.04.86
Revised:

Fetch Material Cross-sections

CALL GFTMAT(IMATE,IPART,MECA,KDIM,TKIN,VALUE,PCUT*,IXST*)*

FETCH and INTERPOLATE the DE/DX and cross-sections tabulated in JMATE banks corresponding to : material IMATE, particle IPART, mechanism name MECA, kinetic energies TKIN.

The MECAnism name can be :

'LOSS' 'PHOT' 'ANNI' 'COMP' 'MUNU' 'BREM' 'PAIR' 'DRAY' 'PFIS' 'HADT' 'HADG'

For Hadronic particles it also computes the total hadronic cross section from TATINA ('HADT') or GHEISHA ('HADG') programs.

Input parameters

IMATE	GEANT material number
IPART	GEANT particle number
MECA	mechanism name of the bank to be fetched
KDIM	dimension of the arrays TKIN, VALUE
TKIN	array of kinetic energies of incident particle (in GeV)

Output parameters

VALUE	array of energy losses (in MeV/cm), or macroscopic cross-sections (in 1/cm)
PCUT(5)	array of the physical cuts for material IMATE (GeV)
IXST	flag = 1 if the array VALUE is filled , = 0 otherwise

Routine called by : < USER > GPLMAT GPRMAT

Note: The common /GCMULO/ contains an array ELOW(90) [CONS 199] with kinetic energy values ranging from 10 KeV to 10 TeV and which can be used as input argument TKIN:
e.g. CALL GFTMAT(10,1,'PHOT',90,ELOW,VALUE,PCUT,IXST)
will return in array VALUE the photo-electric cross-section for a photon in material number 10.

Author(s) : R.Brun, M.Maire
Origin : M.Maire

Submitted: 30.05.86
Revised:

Plot Material Cross-sections

CALL GPLMAT(IMATE,IPART,MECA,KDIM,TKIN,IDM)

INTERPOLATE and PLOT the DE/DX and cross-sections tabulated in JMATE banks corresponding to : material IMATE, particle IPART, mechanism name MECA, kinetic energies TKIN

The MECAnism name can be :

'LOSS' 'PHOT' 'ANNI' 'COMP' 'MUNU' 'BREM' 'PAIR' 'DRAY' 'PFIS' 'HADT' 'HADG' 'ALL '

For Hadronic particles it also computes the total hadronic cross section from TATINA ('HADT') or GHEISHA ('HADG') programs.

Input parameters

IMATE GEANT material number
IPART GEANT particle number
MECA mechanism name of the bank(s) to be printed
if mecan = 'ALL' all the relevant banks for particle IPART will be ploted
KDIM dimension of the array TKIN (maximum 100)
TKIN array of kinetic energies of incident particle (in GeV)
IDM convention for histogramming mode :
IDM.gt.0 fill, print, keep histogram(s)
IDM.eq.0 fill, print, delete histogram(s)
IDM.lt.0 fill, noprint, keep histogram(s)
The histogram IDentifier will be :
10000∗IMATE + 100∗IPART + IMECA
where, IMECA is the link number in structure JMATE
(see CONS 199)
IMECA = 13 for 'HADT', IMECA = 14 for 'HADG'

Called by : < USER >

Author(s) : R.Brun, M.Maire
Origin : M.Maire

Submitted: 30.05.86
Revised:

Print Material Cross-sections

CALL GPRMAT(IMATE,IPART,MECA,KDIM,TKIN)

INTERPOLATE and PRINT the DE/DX and cross-sections tabulated in JMATE banks corresponding to : material IMATE, particle IPART, mechanism name MECA, kinetic energies TKIN

The MECAnism name can be :

'LOSS' 'PHOT' 'ANNI' 'COMP' 'MUNU' 'BREM' 'PAIR' 'DRAY' 'PFIS' 'HADT' 'HADG' 'ALL '

For Hadronic particles it also computes the total hadronic cross section from TATINA ('HADT') or GHEISHA ('HADG') programs.

Input parameters

IMATE	GEANT material number
IPART	GEANT particle number
MECA	mechanism name of the bank(s) to be printed if MECA = 'ALL' all the relevant banks for particle IPART will be printed
KDIM	dimension of the array TKIN (maximum 100)
TKIN	array of kinetic energies of incident particle (in GeV)

Called by : < USER >

Author(s) : R.Brun, M.Maire, J.Allison
Origin : R.Brun, M.Maire

Submitted: 01.06.83
Revised: 09.09.87

Mixtures and Compounds

CALL GSMIXT(IMATE,NAMATE,A,Z,DENS,NLMAT,*WMAT*)

Defines mixture or compound IMATE as composed by the basic NLMAT materials defined through the arrays A, Z and WMAT.

If NLMAT > 0 then WMAT contains the proportion by weights of each basic material in the mixture.

If NLMAT < 0 then WMAT contains the number of atoms of a given kind in the molecule of the compound. In this case, WMAT in output is changed to relative weights.

Mixtures of compounds can also be defined.

IMATE	user material (mixture) number
NAMATE	mixture name (ended by $)
A	array of atomic weights
Z	array of atomic numbers
DENS	density in gr/cm3
NLMAT	see above
WMAT	see above

For a compound the molecular weight and charge are:

$$\mathrm{AMOL} = \Sigma P_iA_i \qquad \mathrm{ZMOL} = \Sigma P_iZ_i \qquad [1]$$

where P_i is the number of atoms of the I-th component in the molecule. (P_i = WMAT(I) in the calling sequence).

Then the proportion by weight is:

$$W_i = P_iA_i/\mathrm{AMOL} \qquad [2]$$

GSMIXT works out an effective atomic weight and atomic number:

$$\mathrm{AEFF} = \Sigma W_iA_i \qquad \mathrm{ZEFF} = \Sigma W_iZ_i \qquad [3]$$

which are stored in the JMATE data structure [CONS 199] together with the radiation length and the absorption length.

The radiation length is computed according to the EGS manual,*

*) EGS manual, SLAC-210 UC-32, June 78, formula 2-6-8 (37).

for an element:

$$\frac{1}{\rho X} = 4\alpha r^2 N \bullet (1/A) \bullet Z(Z+\xi(Z)) \bullet \ [\ \ln(183/Z^{1/3}) - F_c(Z)\ ] \qquad [4]$$

where
- X radiation length (in cm)
- ρ density (in g/cm^3)
- α constant of fine structure
- r classical electron radius (in cm)
- N Avogadro's number
- A atomic weight
- Z atomic number
- $F_c(Z)$ coulomb correction function

$$F_c(Z) = (\alpha Z)^2 \left[\frac{1}{1+(\alpha Z)^2} + 0.20206 - 0.0369(\alpha Z)^2 + 0.0083(\alpha Z)^4 - 0.0020(\alpha Z)^6 \right] \qquad [5]$$

$$\xi(Z) = \ln(1440/Z^{2/3}) \ / \ [\ln(183/Z^{1/3}) - F_c(Z)] \qquad [6]$$

for a compound or mixture:

$$\frac{1}{\rho X} = \Sigma \frac{W_i}{\rho_i X_i} \qquad [7]$$

W_i proportion by weight of the I-th element.

The subroutine GHMIX called by GSMIXT works out an effective atomic weight for a material with NLMAT elements. The criterion is that the hadronic interaction length of a 5 GeV/c pion is correct. Errors on the calculated hadronic interaction length for other momenta and other particles are less than a few % in most cases.

This effective 'hadronic' atomic weight (AHEFF) and the hadronic interaction length are just used within the TATINA package. They are not relevant with the GHEISHA interface.

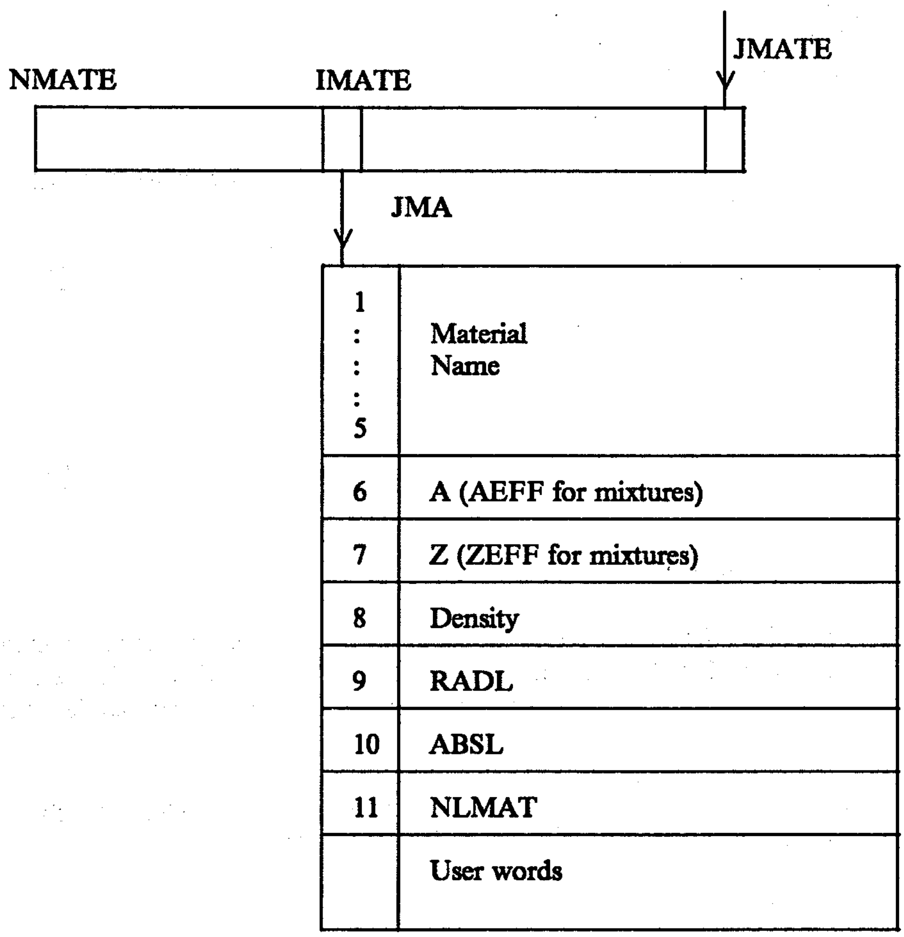
Organisation of the data [CONS 199]

```
JMA = LQ(JMATE-IMATE)
                    NLM = ABS(NLMAT)
JMIXT = LQ(JMA-5)
              Q(MIXT+I) = A(I)
          Q(MIXT+NLM+I) = Z(I)
         Q(MIXT+2NLM+I) = W(I)

JM1 = LQ(JMIXT-1)
               Q(JM1+1) = effective "hadronic" mass: AHEFF (for Tatina)
               Q(JM1+2) = AMOL (compound) or AEFF (mixture)
               Q(JM1+3) = ZMOL (compound) or ZEFF (mixture)
```

Author(s) : R.Brun, M.Maire
Origin : GEANT2/3

Submitted: 01.11.78
Revised: 28.05.86

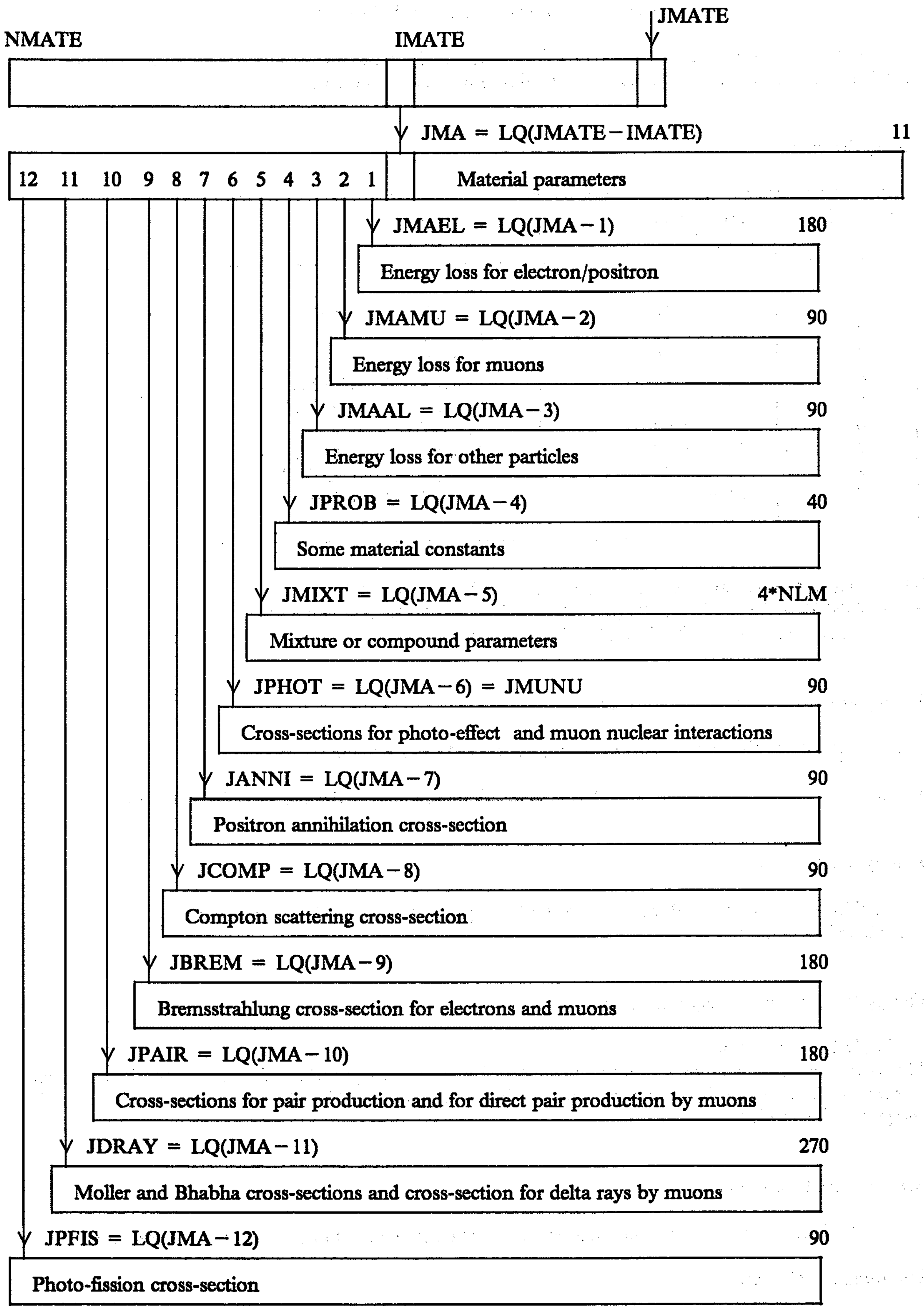
The Material data structure JMATE



1 : : : 5	Material Name
6	A (AEFF for mixtures)
7	Z (ZEFF for mixtures)
8	Density
9	RADL
10	ABSL
11	NLMAT
	User words

JMA = LQ(JMATE−IMATE) pointer to material IMATE

When the subroutine GPHYSI is called at initialisation time the following substructure is created for each material whose number is refered to by any of the user defined tracking media.


JMATE
NMATE
IMATE
JMA = LQ(JMATE−IMATE)
11
12 11 10 9 8 7 6 5 4 3 2 1
Material parameters
JMAEL = LQ(JMA−1)
180
Energy loss for electron/positron
JMAMU = LQ(JMA−2)
90
Energy loss for muons
JMAAL = LQ(JMA−3)
90
Energy loss for other particles
JPROB = LQ(JMA−4)
40
Some material constants
JMIXT = LQ(JMA−5)
4*NLM
Mixture or compound parameters
JPHOT = LQ(JMA−6) = JMUNU
90
Cross-sections for photo-effect and muon nuclear interactions
JANNI = LQ(JMA−7)
90
Positron annihilation cross-section
JCOMP = LQ(JMA−8)
90
Compton scattering cross-section
JBREM = LQ(JMA−9)
180
Bremsstrahlung cross-section for electrons and muons
JPAIR = LQ(JMA−10)
180
Cross-sections for pair production and for direct pair production by muons
JDRAY = LQ(JMA−11)
270
Moller and Bhabha cross-sections and cross-section for delta rays by muons
JPFIS = LQ(JMA−12)
90
Photo-fission cross-section

Comments on contents of the JMATE banks

The energy losses are stored in GeV/cm. The inverse of the macroscopic cross-section, i.e. the mean free path (in cm), is stored instead of the cross-section, for efficiency reasons.

Energy binning

IDECAD	Bin number IEKBIN	Energy range
1	1 -> 10	10 KeV -> 100 KeV
2	11 -> 20	100 KeV -> 1000 KeV
3	21 -> 30	1 MeV -> 10 MeV
4	31 -> 40	10 MeV -> 100 MeV
5	41 -> 50	100 MeV -> 1000 MeV
6	51 -> 60	1 GeV -> 10 GeV
7	61 -> 70	10 GeV -> 100 GeV
8	71 -> 80	100 GeV -> 1000 GeV
9	81 -> 90	1 TeV -> 10 TeV

The values of the bins are kept in the array ELOW(90) in /GCMULO/ :
e.g. ELOW(1) = 10 KeV, ELOW(2) = 20 KeV ... ELOW(10) = 100 KeV
ELOW(11) = 100 KeV, ELOW(12) = 200 KeV ...

Energy loss for electrons and positrons

Words 1 to 90, for electrons: DE/DX = Ionisation (Moller) + Brems.

Words 91 to 180, for positrons: DE/DX = Ionisation (Bhabha) + Brems. [PHYS 330, 340].

Energy loss for muons

DE/DX = Ionisation + Brems. + Direct e^+ e^- production + Nuclear interaction
[PHYS 430, 440, 450].

Energy loss for other charged particles

DE/DX = Ionisation
The values are computed for a proton (mass Mp). For any other particle with mass M and kinetic energy T, one has to compute the equivalent proton kinetic energy as T*Mp/M and look at the corresponding energy binning [PHYS 430].

Some material constants

Various constants which are material dependent and needed to compute the cross-sections.

See routine GPROBI.

Mixture and compound parameters

Words 1 to 4*NLM where NLM is the number of mixture or compound components [CONS 110].

Photo-electric effect. Muon nuclear interaction

As the photo-electric effect vanishes at high energies whereas the muon nuclear interaction cross-section is null at low energies, the two effects are stored within the same bank in order to save space.
From 10 KeV to 50 MeV : Photo-electric effect mean free path [PHYS 230]
From 1 GeV to 10 TeV : Muon nuclear interaction mean free path [PHYS 460].

Positron annihilation

[PHYS 350].

Compton scattering

[PHYS 220].

Bremsstrahlung for electrons and muons

Words 1 to 90, mean free path for electrons [PHYS 340]
Words 91 to 180, mean free path for muons [PHYS 440].

Pair production by gammas and muons

Words 1 to 90, for gammas [PHYS 210].
Words 91 to 180, for muons [PHYS 450}.

Delta ray production by electrons and muons

Words 1 to 90, for $e^-e^- \rightarrow e^-e^-$ [PHYS 330].
Words 91 to 180, for $e^+ e^- \rightarrow e^+ e^-$ [PHYS 330]
Words 181 to 270, for $\mu e^- \rightarrow \mu e^-$ [PHYS 430].

Photo-fission

Only for material with atomic number A > 200 [PHYS 240].

Author(s) : R. Brun
Origin : GEANT2

Submitted: 01.06.83
Revised: 18.08.87

Tracking Medium Parameters and related routines

```
CALL GSTMED(ITMED,NATMED,NMAT,ISVOL,IFIELD,FIELDM,
            TMAXFD,DMAXMS,DEEMAX,EPSIL,STMIN,UBUF,NWBUF)
```

Stores the parameters of the tracking medium ITMED in the data structure JTMED.

ITMED tracking medium number
NATMED tracking medium name (up to 20 characters ended by $)
NMAT material number corresponding to ITMED
ISVOL = 0 if not a sensitive volume
IFIELD = 0 if no magnetic field
= −1 reserved for user decision in GUSWIM
= 1 tracking performed with GRKUTA
= 2 tracking performed with GHELIX
= 3 tracking performed with GHELX3
FIELDM maximum field value (in Kilogauss)
TMAXFD maximum angle due to field permitted in one step (in degrees)
DMAXMS maximum displacement for multiple scattering in one step (in cm)
DEEMAX maximum fractional energy loss in one step
EPSIL tracking precision (in cm)
STMIN minimum step due to energy loss or multiple scattering (in cm)
UBUF array of NWBUF additional parameters
NWBUF

```
CALL GFTMED(ITMED,NATMED*,NMAT*,ISVOL*,IFIELD*,FIELDM*,
            TMAXFD*,DMAXMS*,DEEMAX*,EPSIL*,STMIN*,UBUF*,NWBUF*)
```

Extracts the parameters describing the tracking medium ITMED from the data structure JTMED.

```
CALL GPTMED(ITMED)
```

Prints the tracking medium parameters

ITMED for the tracking medium ITMED.
(for all tracking media if ITMED = 0)

Author(s) : R.Brun
Origin : R.Brun, F.Bruyant

Submitted: 11.02.86
Revised: 23.04.86

Special Tracking Parameters

```
CALL GSTPAR(ITMED,CHPAR,PARVAL)
```

The data structure JTMED contains the standard tracking parameters (CUTS and flags to control the physics processes) which are used by default for all tracking media. It is possible to redefine individually with GSTPAR any of these parameters for a given tracking medium.

ITMED	tracking medium number
CHPAR	character string, name of the variable to be modified
PARVAL	new value (must be given as a floating point).

For example to change CUTGAM to 0.0001 : CALL GSTPAR(ITMED,'CUTGAM',0.0001)

Default parameters	Default	Special parameters for medium JTMED if JTM=LQ(JTMED-ITMED) and JTMN=LQ(JTM)
Q(JTMED+1) = CUTGAM	0.001Gev	Q(JTMN+1) =
+2) = CUTELE	0.001Gev	2
+3) = CUTNEU	0.01Gev	"
+4) = CUTHAD	0.01Gev	"
+5) = CUTMUO	0.01Gev	"
+6) = BCUTE	CUTGAM	"
+7) = BCUTM	CUTGAM	"
+8) = DCUTE	10Tev	"
+9) = DCUTM	10Tev	"
+10) = PPCUTM	0.01Gev	"
+11) = IPAIR	2	"
+12) = ICOMP	2	"
+13) = IPHOT	2	"
+14) = IPFIS	0	"
+15) = IDRAY	0	"
+16) = IANNI	2	"
+17) = IBREM	2	"
+18) = IHADR	2	"
+19) = IMUNU	0	"
+20) = IDCAY	2	"
+21) = ILOSS	2	"
+22) = IMULS	1	+22
26+29) = GHEISHA		see PHYS700

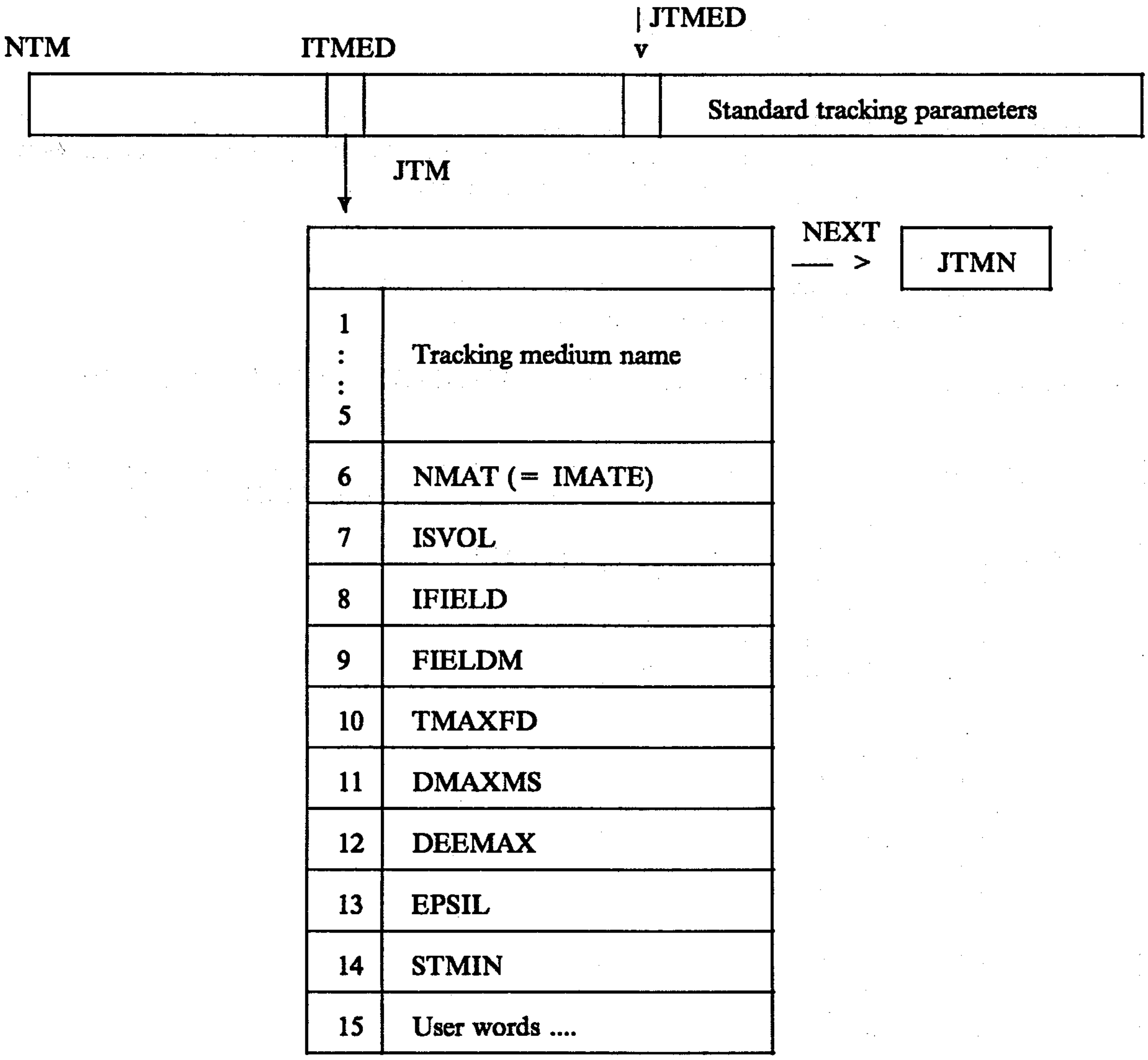
For comments on default values for IDRAY and ILOSS see [BASE 040].

Note: At tracking time the parameters above are copied from JTMED or JTMN into the COMMON blocks /GCCUTS/ and /GCPHYS/

Author(s) :
Origin : GEANT2/3

Submitted: 01.11.78
Revised: 28.05.86

The Tracking Medium data structure JTMED



1 : : 5	Tracking medium name
6	NMAT (= IMATE)
7	ISVOL
8	IFIELD
9	FIELDM
10	TMAXFD
11	DMAXMS
12	DEEMAX
13	EPSIL
14	STMIN
15	User words ....

JTM = LQ(JTMED−ITMED) pointer to tracking medium ITMED
JTMED contains the standard parameters
JTMN = LQ(JTM) contains the tracking parameters as modified by GSTPAR.
[CONS 210]

Author(s) : R.Brun, G.N.Patrick
Origin : GEANT2

Submitted: 01.06.83
Revised: 14.04.87

Standard Particle definition and related routines

CALL GPART

Stores the standard particle constants in the data structure JPART and, through the routine GSDK, their decay modes [CONS310].

Particle	No.	Mass(GeV)	Charge	Life time(sec)
Gamma	1	.0	0.	stable (10^{15})
Positron	2	.000511	1.	stable
Electron	3	.000511	-1.	stable
Neutrino	4	.0	0.	stable
Muon +	5	.105659	1.	$2.19709*10^{-6}$
Muon -	6	.105659	-1.	$2.19709*10^{-6}$
Pion 0	7	.134693	0.	$8.3*10^{-17}$
Pion +	8	.139567	1.	$2.603*10^{-8}$
Pion -	9	.139567	-1.	$2.603*10^{-8}$
Kaon 0 long	10	.49767	0.	$5.183*10^{-8}$
Kaon +	11	.493667	1.	$1.2371*10^{-8}$
Kaon -	12	.493667	-1.	$1.2371*10^{-8}$
Neutron	13	.939573	0.	898
Proton	14	.93828	1.	stable
Antiproton	15	.93828	-1.	stable
Kaon 0 short	16	0.49767	0.	$8.923*10^{-11}$
Eta	17	0.5488	0.	$7.479742*10^{-19}$
Lambda	18	1.11560	0.	$2.632*10^{-10}$
Sigma +	19	1.18936	1.	$8.00*10^{-11}$
Sigma 0	20	1.19246	0.	$5.80*10^{-20}$
Sigma -	21	1.19734	-1.	$1.482*10^{-10}$
Xi 0	22	1.31490	0.	$2.90*10^{-10}$
Xi -	23	1.32132	-1.	$1.641*10^{-10}$
Omega	24	1.67245	-1.	$8.19*10^{-11}$
Antineutron	25	0.939573	0.	898
Antilambda	26	1.11560	0.	$2.632*10^{-10}$
Antisigma -	27	1.18936	-1.	$8.0*10^{-11}$
Antisigma 0	28	1.19246	0.	$5.8*10^{-20}$
Antisigma +	29	1.19734	1.	$1.482*10^{-10}$
Antixi 0	30	1.3149	0.	$2.9*10^{-10}$
Antixi +	31	1.32132	1.	$1.641*10^{-10}$
Antiomega +	32	1.67245	1.	$8.19*10^{-10}$

(see next page -->)

Particle	No.	Mass	Charge	Life time
Tau +	33	1.7842	1.	$3.4*10^{-13}$
Tau -	34	1.7842	1.	$3.4*10^{-13}$
D+	35	1.8694	1.	$9.2*10^{-13}$
D-	36	1.8694	-1.	$9.2*10^{-13}$
D^0	37	1.8647	0.	$4.4*10^{-13}$
Anti D^0	38	1.8647	0.	$4.4*10^{-13}$
F+	39	1.9710	1.	$1.9*10^{-13}$
F-	40	1.9710	-1.	$1.9*10^{-13}$
Lambda C+	41	2.2822	1.	$2.3*10^{-13}$
W+	42	80.8000	1.	$9.4*10^{-26}$
W-	43	80.8000	-1.	$9.4*10^{-26}$
Z^0	44	92.9000	0.	$7.74*10^{-26}$
Deuteron	45	1.877	1.	stable
Tritium	46	2.817	1.	stable
Alpha	47	3.755	2.	stable
Geantino	48	0.	0.	stable

Note If the user does not need all the above particles, or needs more particles, or wants to partly override the standard, the routines GSPART and GSD can be called instead to define particles one at a time.

All data taken from M. Aguilar-Benitez.

For the W and Z, the lifetimes are calculated from quoted upper limits on widths.

CALL GSPART(IPART,NAPART,ITRTYP,AMASS,CHARGE,TLIFE,UB,NWB)

Stores the constants describing the particle IPART in the data structure JPART.

IPART particle number

NAPART particle name (up to 20 characters ended by $)

ITRTYP type of tracking routine requested
ITRTYP = 1 particle tracked by GTGAMA
ITRTYP = 2 particle tracked by GTELEC
ITRTYP = 3 particle tracked by GTNEUT
ITRTYP = 4 particle tracked by GTHADR
ITRTYP = 5 particle tracked by GTMUON
ITRTYP = 6 geantino tracked by GTNINO

AMASS particle mass

CHARGE particle charge

TLIFE particle life time (in seconds)

UB array of NWB user additional parameters

NWB

*) M. Aguilar-Benitez et al, Review of Particle Properties, Rev. Mod. Phys. 56(198

CALL GFPART(IPART,NAPART,ITRTYP*,AMASS*,CHARGE*,TLIFE*,UB*,NWB*)*

Extracts the constants describing the particle IPART from the data structure JPART.

CALL GPPART(IPART)

Prints the particle constants

IPART for the particle IPART
(for all particles if IPART = 0)

GEANT 3.10 USER'S GUIDE CONS 310

Author(s) : G.N.Patrick
Origin : R.Brun, G.N.Patrick

Submitted: 23.01.84
Revised: 30.05.86

Branching Ratios and Particle Decay Modes

```
CALL GSDK(IPART,BRATIO,MODE)
```

GSDK stores the branching ratios and partial decay modes for two and three-body particle decays into the data structure JPART.

IPART	GEANT particle number
BRATIO(6)	Up to six branching ratios (%)
MODE(6)	Up to six partial decay modes

The decay modes should be coded into the array MODE such that:

$$\mathrm{MODE(I)} = \mathrm{N3}*10^4 + \mathrm{N2}*10^2 + \mathrm{N1} \quad \text{for } \mathrm{I} = 1,\dots,6$$

where,

N1 = GEANT particle number for decay product 1
N2 = GEANT particle number for decay product 2
N3 = GEANT particle number for decay product 3 (if any)

It is important to note the following:

a) Prior to calling GSDK, all parent and secondary particles should have been defined by a previous call to GSPART.

b) If less than six decay modes are defined the remaining elements of BRATIO and MODE must be preset to zero.

For a given particle, IPART, the decay parameters are stored into the JPART data structure as follows:

JPA = LQ(JPART−IPART)	pointer to particle IPART
JDK1 = LQ(JPA−1)	pointer to b.ratio banks
JDK2 = LQ(JPA−2)	pointer to decay mode banks
BR(I) = Q(JDK1+I)	Ith. branching ratio
MODE(I) = IQ(JDK2+I)	Ith. decay mode, where I=1,...,6.

When a non stable particle decays during the tracking, the routine GDECAY is called. If the decay modes and branching ratios have been stored by GSDK, then GDECAY generates the decay products in the 2- or 3-body phase space. For any other particle the user routine GUDCAY is called, where the user is free to code special decay modes and branching ratios.

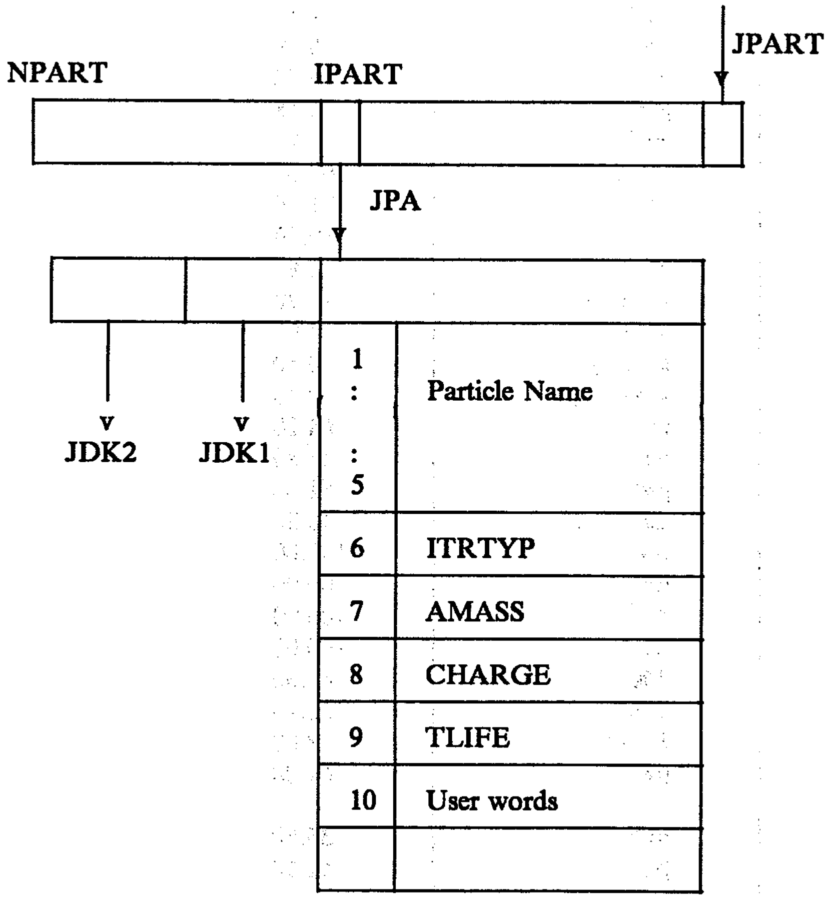
All data taken from M. Aguilar-Benitez et al., Review of Particle Properties, Rev. Mod. Phys. 56(1984).

Parent Particle(s)	Decay	Branching Ratio (%)
μ^+,μ^-	$e\nu\nu$	100.00
π^0	$\gamma\gamma$	98.802
	$\gamma e^+ e^-$	1.198
π^+,π^-	$\mu\nu$	100.00
K^0_L	$\pi e\nu$	38.70
	$\pi\mu\nu$	27.10
	$\pi^0\pi^0\pi^0$	21.50
	$\pi^+\pi^-\pi^0$	12.39
K^+,K^-	$\mu\nu$	63.51
	$\pi\pi^0$	21.17
	$\pi\pi^+\pi^-$	5.59
	$e\nu\pi^0$	4.82
	$\mu\nu\pi^0$	3.18
	$\pi\pi^0\pi^0$	1.73
$K^0{}_s$	$\pi^+\pi^-$	68.61
	$\pi^0\pi^0$	31.39
η	$\gamma\gamma$	39.00
	$\pi^0\pi^0\pi^0$	31.80
	$\pi^+\pi^-\pi^0$	23.70
	$\pi^+\pi^-\gamma$	4.91
	$e^+e^-\gamma$	0.50
	$\pi^0\gamma\gamma$	0.10
Λ	$p\pi^-$	64.20
	$n\pi^0$	35.80
Σ^+	$p\pi^0$	51.64
	$n\pi^+$	48.36
Σ^0	$\Lambda\gamma$	100.00
Σ^-	$n\pi^-$	100.00
Ξ^0	$\Lambda\pi^0$	100.00
Ξ^-	$\Lambda\pi^-$	100.00
Ω^-	ΛK^-	68.60
	$\Xi^0\pi^-$	23.40
	$\Xi^-\pi^0$	8.00
$\bar{\Lambda}$	$\bar{p}\pi^+$	64.20
	$\bar{n}\pi^0$	35.80
$\bar{\Sigma}^-$	$\bar{p}\pi^0$	51.64
	$\bar{n}\pi^-$	48.36
$\bar{\Sigma}^0$	$\bar{\Lambda}\gamma$	100.00
$\bar{\Sigma}^+$	$\bar{n}\pi^+$	100.00
$\bar{\Xi}^0$	$\bar{\Lambda}\pi^0$	100.00
$\bar{\Xi}^+$	$\bar{\Lambda}\pi^+$	100.00
$\bar{\Omega}^+$	$\bar{\Lambda}K^+$	68.60
	$\bar{\Xi}^0\pi^+$	23.40
	$\bar{\Xi}^+\pi^0$	8.00

Author(s) :
Origin : GEANT2/3

Submitted: 01.11.78
Revised: 28.05.86

The Particle data structure JPART



JPA = LQ(JPART − IPART) pointer to particle number IPART
JDK1 = LQ(JPA − 1) pointer to branching ratio banks
JDK2 = LQ(JPA − 2) pointer to decay mode banks
Q(JDK1 + I) = Ith branching ratio I = 1,6
Q(JDK2 + I) = Ith decay mode I = 1,6

[CONS 310]

Author(s) : P.Zanarini
Origin : R.Brun, P.Zanarini

Submitted: 01.10.84
Revised: 14.05.86

Introduction to the Drawing package

1. The drawing package

The drawing package has been designed mainly to:

- draw the detector
- draw the detector geometrical tree
- draw the detector geometrical specification
- draw particle trajectories
- draw hits.

2. Drawing the detector

The detector can be looked at from any view point and with any scale factor (GDRAW, GDRVOL); appropriate attributes can be set in order to see only selected objects (so avoiding messy pictures); hidden line removal is an option that should be added to the package later on.

'Cut' views, i.e. intersections of a given plane with the detector, can also be displayed (GDRAWC, GDRAWX). This feature is very useful to see internal details that the routine GDRAW would not show.

3. Drawing the geometrical tree

The geometrical tree (GDTREE) is a representation of the overall structure of the detector, namely the mother-daughter relationship among the various objects composing it. Several types of additional information are available on request: multiplicity of a given volume (i.e. how many times it is positioned in other places, or number of subdivisions), detector nature, visibility flag, etc. This drawing tree capability can be very useful when tuning the detector geometry.

4. Drawing the geometrical specifications

The geometrical specifications (GDSPEC) give a detailed picture of one particular piece of the detector. Three drawings of the volume (a projection view and two cut views), its shape type and numerical parameters (i.e. dimensions), and a scale to aid geometric calculations, are presented together in a single graphics frame. The set of geometrical specifications of all the descendants of a given node on the tree, can easily be obtained each on a separate picture with the routine GDFSPC.

5. Drawing particle trajectories

The tracks generated by the tracking package, and stored in the data structure JXYZ, can be easily drawn with the routine GDXYZ.

The names of the particles and/or the track numbers can be drawn as well (GDPART).

Four types of representations are used to display the classes of particles, with different colours and line styles:

i) red solid lines for charged particles (GTELEC, GTHADR)

ii) green dashed lines for muons (GTMUON)

iii) black blank/dotted lines for neutral particles (GTNEUT)

iv) blue dotted lines for gammas (GTGAMA)

A special routine has been provided to display the tracks online (GDCXYZ), that could be called for instance by GUSTEP. That routine shows the tracks exactly at the same time they are generated by the tracking package of GEANT3, giving so a useful interactive debugging tool.

6. Drawing hits

The hits generated by the tracking package and stored in the data structure JHITS, can be displayed by the hits routines, with different functionality:

i) draw one hit (GDAHIT); called by user routines

ii) draw all the hits of trajectory type sets/detectors (GDHITS)

iii) draw all the hits of calorimeter type sets/detectors (GDCHIT)

Different symbols for every subdetector can be used, chosen among hardware characters (dots), software crosses, or from the HPLOT table of software characters. The size of the software characters and crosses is given as an argument to GDAHIT/GDHITS, while it is computed as a function of the hits value in GDCHIT.

7. The view banks

The basic detector drawing routines (GDRAW, GDRAWC, GDRAWX) have to scan the data structure JVOLUM repeatedly. When the detector is described in a very detailed way, the time spent in the interpretation of the JVOLUM bank and in the 3D transformations can increase dramatically. For a detector with more than 100 different volume names, for example, this time can reach 3-4 real time minutes on a VAX 780, whereas the specific time required just by the drawing would be only a few seconds.

In order to avoid this problem the 'bank-mode' routines have been developed. The basic idea is to separate the interpretation (i.e. the JVOLUM scanning to convert the 3D geometry structure into a set of 2D lines) from the drawing itself. In this way, the interpretation is performed only once and all the 2D information is stored in view banks (data structure JDRAW) [DRAW 399]. These views can then be looked at in a quicker way, having only to draw all 2D vectors previously stored. For a detector with more than 100 different volume names, for example, this is achieved at the cost of only a few thousand words of memory for each view bank.

One can therefore open a view bank (GDOPEN), identified by a number, perform appropriate drawings (only interpretation will be made, of course), close the bank (GDCLOS) and finally look at the picture stored in it (GDSHOW). When a view bank has been closed it cannot be modified anymore, but it can be displayed as many times as wanted (GDSHOW) or deleted (GDELET).

8. Other features

The user can control some drawing options (GDOPT), by selecting for instance to have either parallel or perspective projection, either Y-Z or R-Z projection, etc.

There is a routine (GDZOOM) that, if called, applies a zoom to everything (volumes, tracks, hits, etc.) that will be drawn from then on. This feature, in conjunction with the interactive command MEASURE [XINT 110], can be used for detailed viewing.

Another tool that could help in the interactive debugging or tuning of the detector geometry is the interactive command EDITV [XINT 130], by which it is possible to modify interactively some geometrical parameters set by the user routines defining the detector geometry.

It is possible to draw the axes of the 3D MAster Reference System (GDAXIS) oriented in agreement with the current view point.

Two other routines draw a scale (GDSCAL) or a profile of a man (GDMAN) in 2D user coordinates to give an idea of the dimensions within current scale factors.

A 2D text (GDRAWT) using software characters (hardware characters should be implemented later on), 2D vectors (GDRAWV) or a frame header (GDHEAD) are also available.

Attributes like colour (GDCOL) and line width (GDLW) can be globally set for all 2D drawings (i.e. text, vectors, man, etc.); such attributes do not affect the volume attributes that can be set by the GSATT routine with 'COLO' or 'LWID' option.

A graphics input is available (GDCURS) to fetch the 2D user coordinates of the graphics cursor on the screen, allowing an immediate user interface with the interactive version of GEANT3. In particular there are interactive commands to zoom, measure, pick tracks or hits points that make use of that routine.

Various conversions from 3D to 3D, and 3D to 2D coordinates are performed by GDFR3D and GD3D3D.

Edition of various geometrical parameters of an existing volume can be made through the use of the routine GEDITV.

9. Basic and advanced graphics

On the low-end graphics, either Mini-GD3[1] (using PIGS), GD3[2] , or GKS[3] may be +USE selected when preparing the GEANT3 program. PIGS or GKS are used by the interactive version of GEANT3, while GD3 is used on the IBM/MVS system and on the CDC at CERN.

On the high-end graphics, an interface to PIONS[4] allows dynamic 3D graphics. The GPIONS patch of GEANX pam has to be loaded in addition to GEANT3 in order to activate the PIONS interface.

Another interface exists with the Apollo Graphics Metafile Resource GMR-3D. The GGMR

[1] Mini-GD3 Introductory guide for users - CERN/DD/US/90

[2] GD3 Introductory guide for users on the IBM - CERN/DD/US/27

[3] Guide to the use of GKS at CERN - CERN/DD/EE/85-3

[4] The PIONS User Guide - CERN/DD/EE/85-1
PIONS, A High Performance Graphics Package - CERN/DD/EE/85-2

patch of GEANX pam has to be loaded in addition to GEANT3 in order to activate the GMR interface.

10. Running instructions

On CERN VAX's the metafiles produced by GKS can be interpreted and sent to the Versatec electrostatic plotters. The procedure GKSVT may be used on VAX-8600. The capture files produced by PIGS can be replayed online on a graphics terminal by calling the PIGS routine CPREP. An updated version of CPREP supporting colours (with the main program REP), as well as a procedure to make hard copy of the PIGS capture files on the Versatec electrostatic plotters, are available on request to the GEANT3 team.

On the CERN IBM/MVS system, GD3 produces a metafile on the unit specified in the CALL TVBGN in the main program [BASE 100]. Hard copy of the GD3 metafiles may be produced on the Versatec electrostatic plotters via the GD3VT procedure.

This procedure, when selecting the large Versatec plotter at CERN (via the key WP), uses a plot parameter file that performs an automatic scaling by factor 3 of the user coordinates. Therefore, without changing anything in its program, the user will get a picture 3 times bigger (on a 3 times larger paper) just by selecting the WP parameter.

11. Summary

The drawing package is initialized by (in the order):

TVBGN	to initialize the PIGS, GKS, or GD3 basic graphics package
GDINIT	to initialize the drawing package

Main drawing routines are:

GDRAW	to draw a projection view of the detector - Case 1
GDRVOL	to draw a projection view of the detector - Case 2
GDRAWC	to draw a cut view of the detector (along one axis)
GDRAWX	to draw a cut view of the detector (from any point)
GDXYZ	to draw tracks at the end of the event
GDCXYZ	to draw tracks at tracking time
GDPART	to draw particle names and track numbers at end of tracks
GDAHIT	to draw one single hit
GDHITS	to draw hits for 'trajectory' type detectors
GDCHIT	to draw hits for 'calorimeter' type detectors

Routines that show how the detector has been modeled are:

GDTREE	to draw a picture with the geometrical tree
GDSPEC	to draw a picture with volume specifications
GDFSPC	to draw several GDSPEC pictures

Routines that perform control operations on view banks are:

GDOPEN	to open a given view bank, identified by a view number; in this way we enter in bank-mode
GDCLOS	to close current view bank, i.e. the last one opened, and restore screen-mode
GDSHOW	to show all graphics information contained in a given view bank
GDELET	to delete a given view bank from memory

Other routines are:

GDOPT	to set drawing options
GDZOOM	to set the zoom parameters
GDAXIS	to draw the axes of the MARS, oriented according to the current view parameters
GDSCAL	to draw the current scale
GDMAN	to draw a profile of a man within the current scale
GDRAWT	to draw text, with software characters
GDRAWV	to draw polylines in 2D user coordinates
GDHEAD	to draw a frame header
GDCOL	to set colour code
GDLW	to set line width
GDCURS	to have an input from the graphics cursor
GDFR3D	to convert from 3D coordinates (either in MARS or DRS) to 2D user coordinates
GD3D3D	to convert from 3D MARS coordinates to 3D Projection Reference System coordinates.

Author(s) : R.Brun, A.McPherson, P.Zanarini
Origin : Same

Submitted: 15.08.83
Revised: 20.05.86

Drawing a Volume Projection view – Case 1

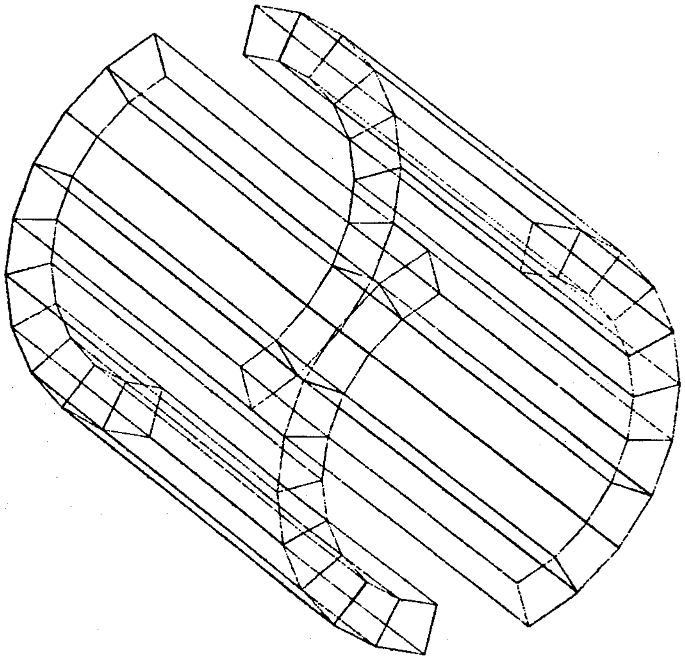
```
CALL GDRAW (NAME,THETA,PHI,PSI,U0,V0,SU,SV)
```

Draws an orthographic parallel projection or a perspective projection (depending on the option chosen by GDOPT) of the volume NAME with all its descendants, at the position U0,V0 (user coordinates), with the scale factors SU and SV; the object is seen from THETA and PHI angles, and the resulting 2D projection is also rotated by an angle PSI on the screen plane. These parameters, as well as zoom parameters set by GDZOOM, define the current 'view parameters', and they are copied in /GCDRAW/.

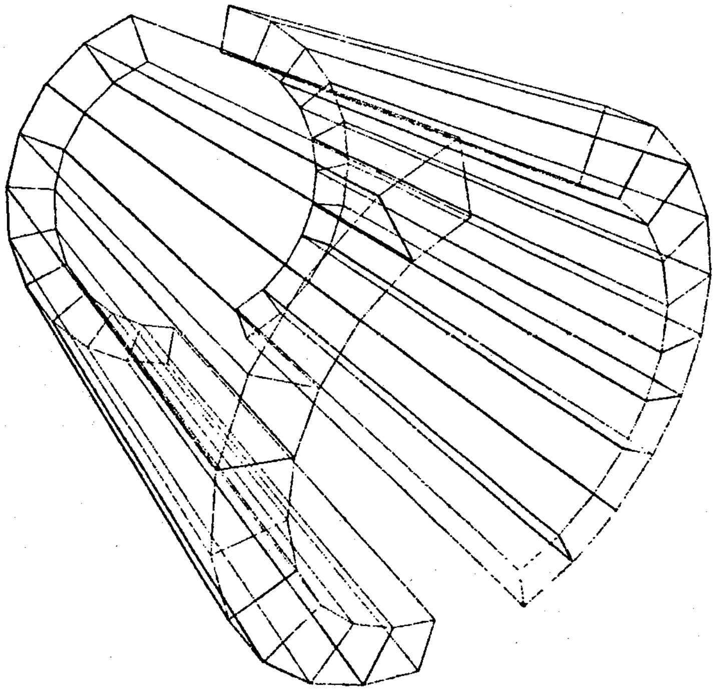
Attributes like colour, fill area, line width, line style, visibility, etc. can be set before, by calling the GSATT routine for NAME and/or its descendants [GEOM 500].

NAME	volume name
THETA	theta angle between the line of sight and the Z-axis of MAster Reference System
PHI	phi angle between the projection of the line of sight on plane X-Y and the X-axis of MARS
PSI	psi angle by which the projected image will be rotated on the screen plane
U0	u-coordinate on the screen of the volume origin
V0	v-coordinate on the screen of the volume origin
SU	scale factor for u-coordinates
SV	scale factor for v-coordinates

```
CALL GDRAW(NAME,45.,135.,0.,10.,10.,0.01,0.01)
```



```
CALL GDOPT('PROJ','PERS')
CALL GDRAW(NAME,45.,135.,0.,10.,10.,0.01,0.01)
```



Author(s) : P.Zanarini
Origin : Same

Submitted: 01.01.86
Revised: 20.05.86

Drawing a Volume Projection view – Case 2

```
CALL GDRVOL (N,LNAMES,LNUMBS,NRS,THETA,PHI,PSI,U0,V0,SU,SV)
```

Draws an orthographic parallel projection or a perspective projection (depending on the option chosen by GDOPT) of the volume LNAMES(N),LNUMBS(N) with all its descendants, at the position U0,V0 (user coordinates), with the scale factors SU and SV; the object is seen from THETA and PHI angles, and the resulting 2D projection is also rotated by an angle PSI on the screen plane. These parameters, as well as zoom parameters set by GDZOOM, define the current 'view parameters', and they are copied in /GCDRAW/.

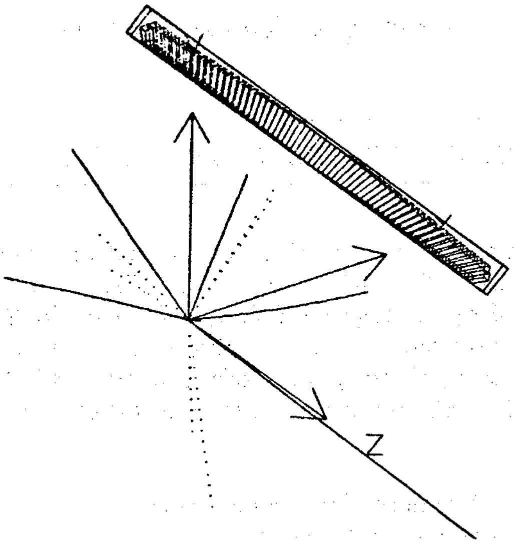
Attributes like colour, fill area, line width, line style, visibility, etc. can be set before, by calling the GSATT routine for LNAMES(N) and/or its descendants [GEOM 500].

This routine differs from GDRAW in the following :

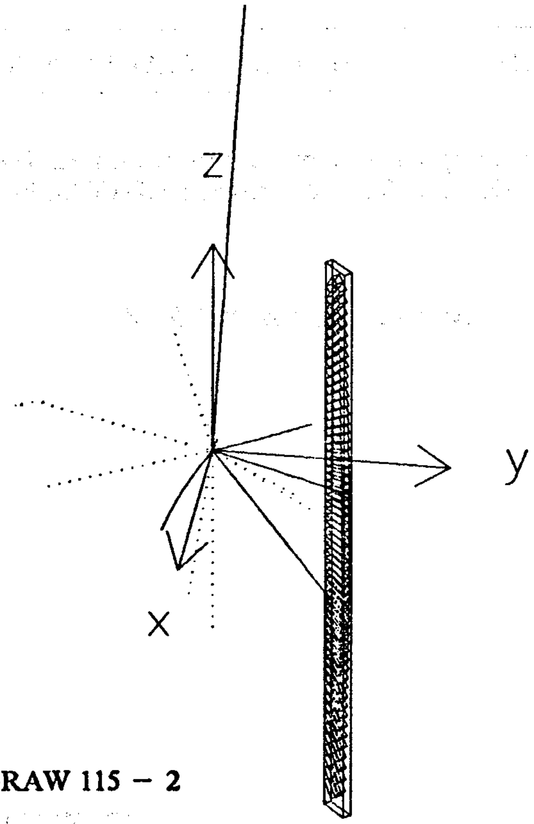
- the object to be drawn is identified by a full path, giving so the possibility to draw a particular copy or division of a volume, or even a volume that has more than one mother in the geometry tree; LNAMES(1),...,LNAMES(N) contain the volume names and LNUMBS(1),...,LNUMBS(N) the volume numbers defining the path to go from the top volume to the one to be drawn
- the object can be drawn either with respect to the MAster Reference System (NRS = 0) or with respect to its Daugther Reference System (i.e. the Local R.S.); in the first case it is drawn where it stands in the real world, while in the second one it is drawn like GDRAW would do
- in this latter case, track and hit points will be drawn with respect to the DRS of the volume last drawn by this routine, and not with respect to the MARS as it done normally (to reset to the normal case a call with NRS = 0 or N = 0 is required)

N	number of levels in the arrays LNAMES,LNUMBS LNAMES(N),LNUMBS(N) (i.e. the bottom volume of this path) is also the one that is actually drawn
LNAMES	array of volume name (dimensioned at least to N)
LNUMBS	array of volume numbers (dimensioned at least to N)
NRS	reference system used : NRS = 0 to have the volume(s) drawn with respect to the MARS NRS < > 0 to have the volume(s) drawn with respect to the DRS
THETA	theta angle between the line of sight and the Z-axis of MAster Reference System
PHI	phi angle between the projection of the line of sight on plane X-Y and the X-axis of MARS
PSI	psi angle by which the projected image will be rotated on the screen plane
U0	u-coordinate on the screen of the volume origin
V0	v-coordinate on the screen of the volume origin
SU	scale factor for u-coordinates
SV	scale factor for v-coordinates

```
DIMENSION LNAMES(5),LNUMBS(5)
DATA LNAMES/'OPAL','BRL-','EB  ','EBB ','EBP '/
DATA LNUMBS/    1 ,    1 ,    1 ,    1 ,   20 /
- - - - -

NRS=0
CALL GDRVOL(5,LNAMES,LNUMBS,NRS,45.,135.,0.,10.,10.,0.01,0.01)
CALL GDAXIS(0.,0.,0.,300.)
CALL GDXYZ(0)
```



```
- - - - -

NRS=1
CALL GDRVOL(5,LNAMES,LNUMBS,NRS,45.,135.,0.,10.,10.,0.01,0.01)
CALL GDAXIS(0.,0.,0.,300.)
CALL GDXYZ(0)
```



Author(s) : P.Zanarini
Origin : Same

Submitted: 15.05.84
Revised: 20.05.86

Drawing a Volume Cut view

```
CALL GDRAWX(NAME,CUTTHE,CUTPHI,CUTVAL,THETA,PHI,U0,V0,SU,SV)
```

Draws a 'cut view' of the volume NAME, with all its descendants, i.e. draws their intersection with the cut plane normal to the line given by the angles CUTTHE, CUTPHI at the distance CUTVAL from the origin. The view point is defined by the angles THETA, PHI. U0, V0, SU, SV have the same meaning as in GDRAW. These 'view parameters', as well as zoom parameters set by GDZOOM, are copied in /GCDRAW/. Attributes like colour, fill area, line width, line style, visibility, etc. can be set before, by calling the GSATT routine for NAME and/or its descendants [GEOM 500].

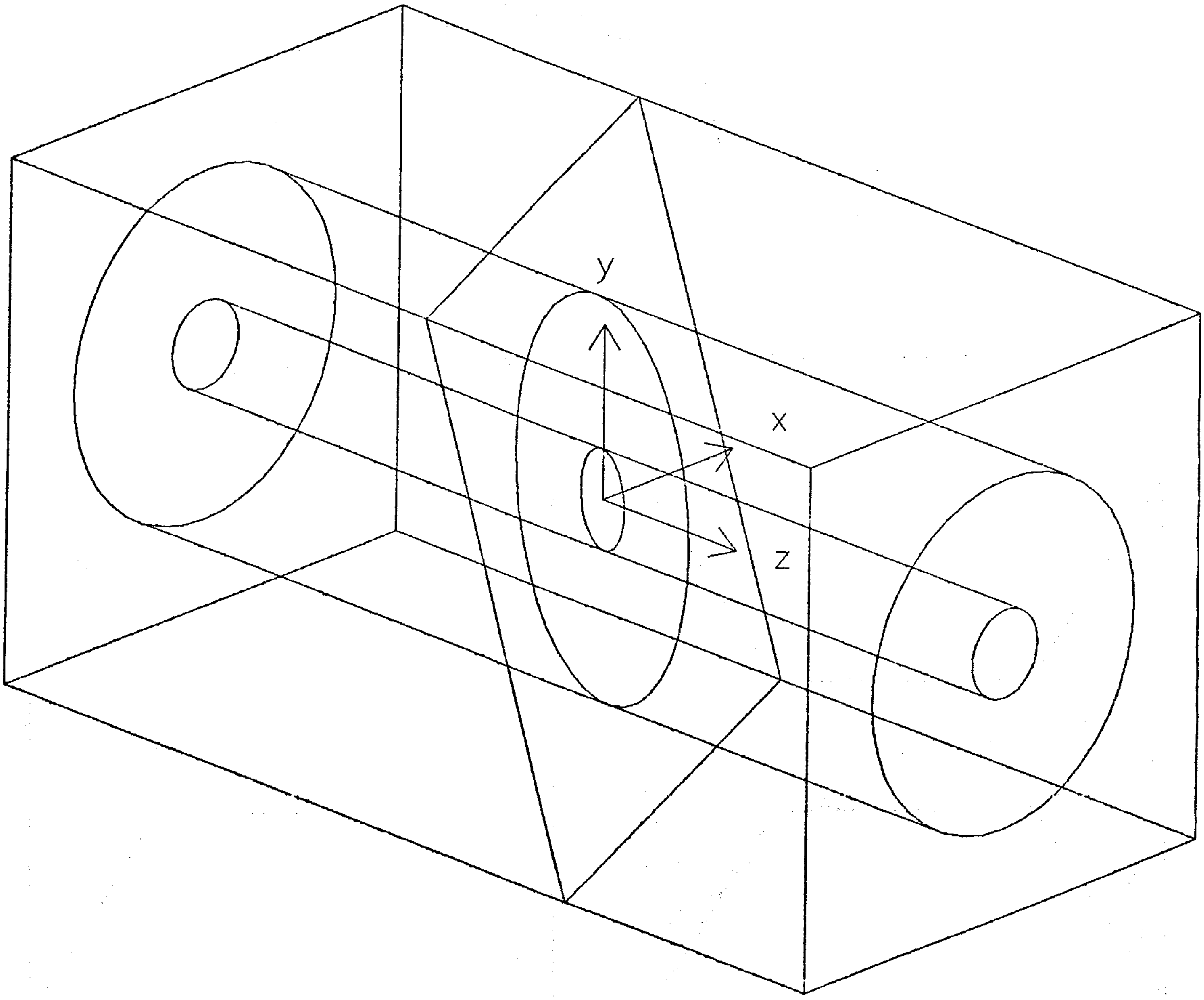
CUTTHE	theta angle of the line normal to the cut plane
CUTPHI	phi angle of the line normal to the cut plane
THETA	viewing angle theta
PHI	viewing angle phi
U0	u-coordinate on the screen of the volume origin
V0	v-coordinate on the screen of the volume origin
SU	scale factor for u-coordinates
SV	scale factor for v-coordinates

```
CALL GDRAWC(NAME,IAX,CUTVAL,U0,V0,SU,SV)
```

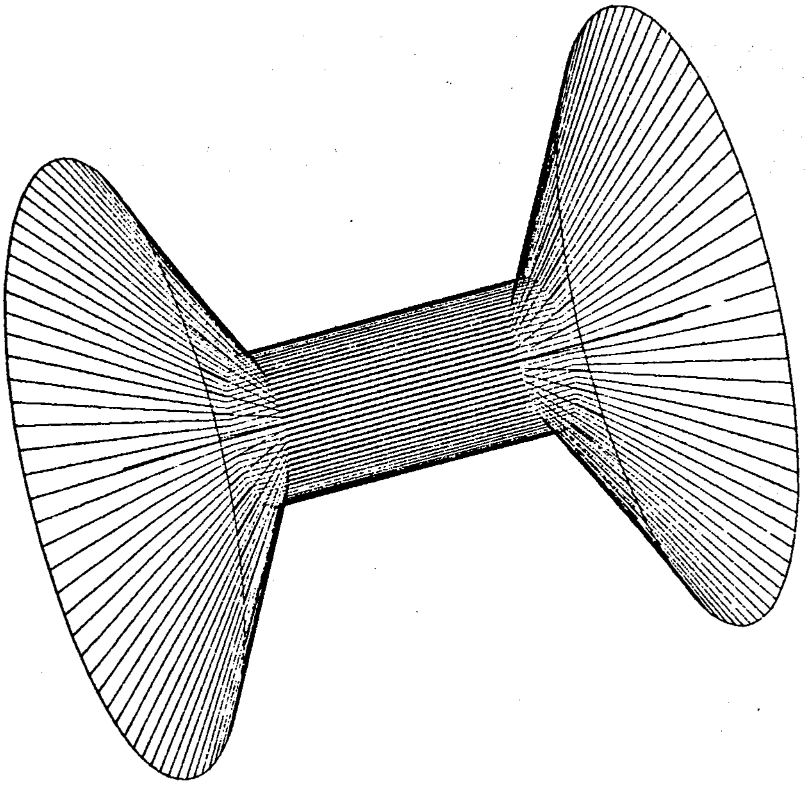
This routine is a special case of the previous GDRAWX. Now the cut plane is normal to one of the main axes (IAX) and placed at a distance CUTVAL from the origin. The view point is along the same axis.

IAX	axis : 1 for X, 2 for Y, 3 for Z

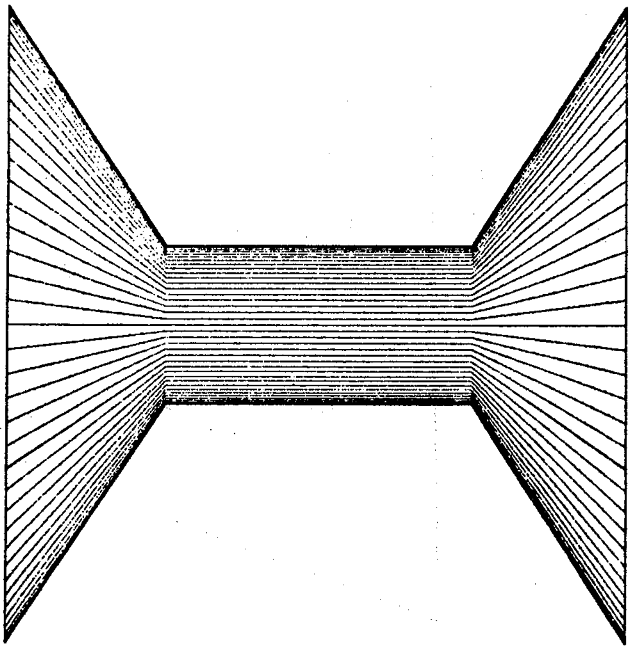
```
CALL GDAXIS(0.,0.,0.,200.)

CALL GDRAW(NBOX,50.,150.,0.,10.,10.,0.01,0.01)
CALL GDRAW(NTUBE,50.,150.,0.,10.,10.,0.01,0.01)

CALL GDRAWX(NBOX,30.,40.,0.,50.,150.,10.,10.,0.1,0.1)
CALL GDRAWX(NTUBE,30.,40.,0.,50.,150.,10.,10.,0.1,0.1)
```



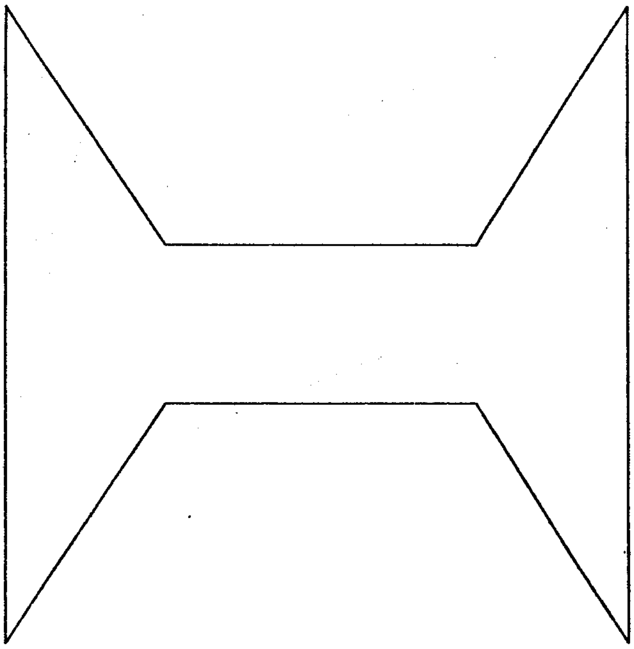
CALL GDRAW(NAME,60.,30., ...)

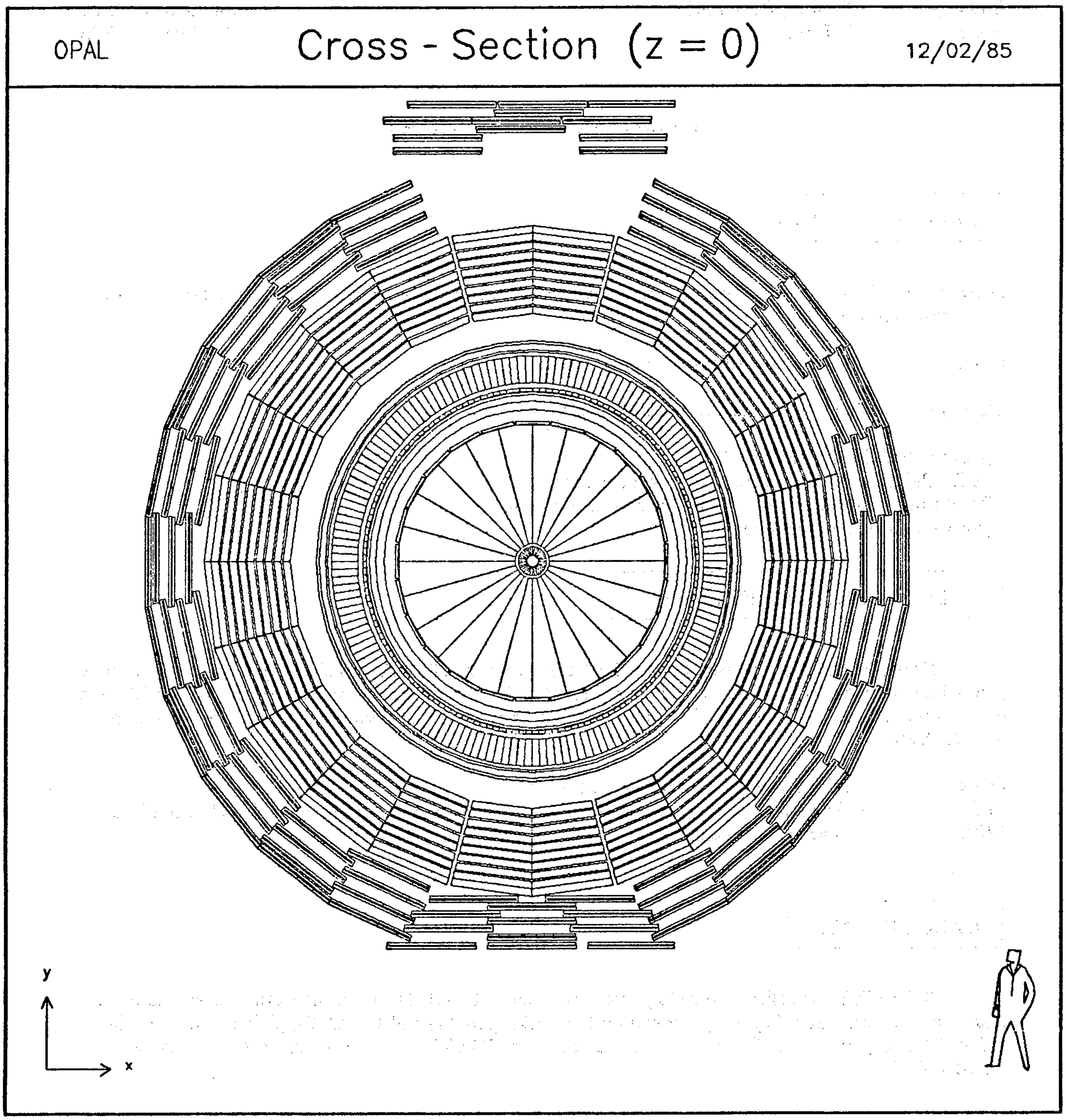


CALL GDRAW(NAME,90.,180., ...)



CALL GDRAWC(NAME,1,0., ...)





OPAL Cross - Section (z = 0) 12/02/85

Author(s) : R.Brun, P.Zanarini
Origin : R.Brun

Submitted: 15.05.84
Revised: 20.05.86

Drawing Particle Trajectories

CALL GDXYZ(ITRA)

Draws track ITRA for which space points have been stored in bank JXYZ. The view parameters are taken from /GCDRAW/.

ITRA track number (if 0 all tracks are taken)

The colour and line style corresponds to the track type :

(blue) dotted line for gammas	....
(red) solid line for charged particles (except muons)	________
(black) blank/dotted line for neutral hadrons or neutrinos	. . . .
(green) dashed line for muons	-----------

CALL GDPART(ITRA,ISEL,SIZE)

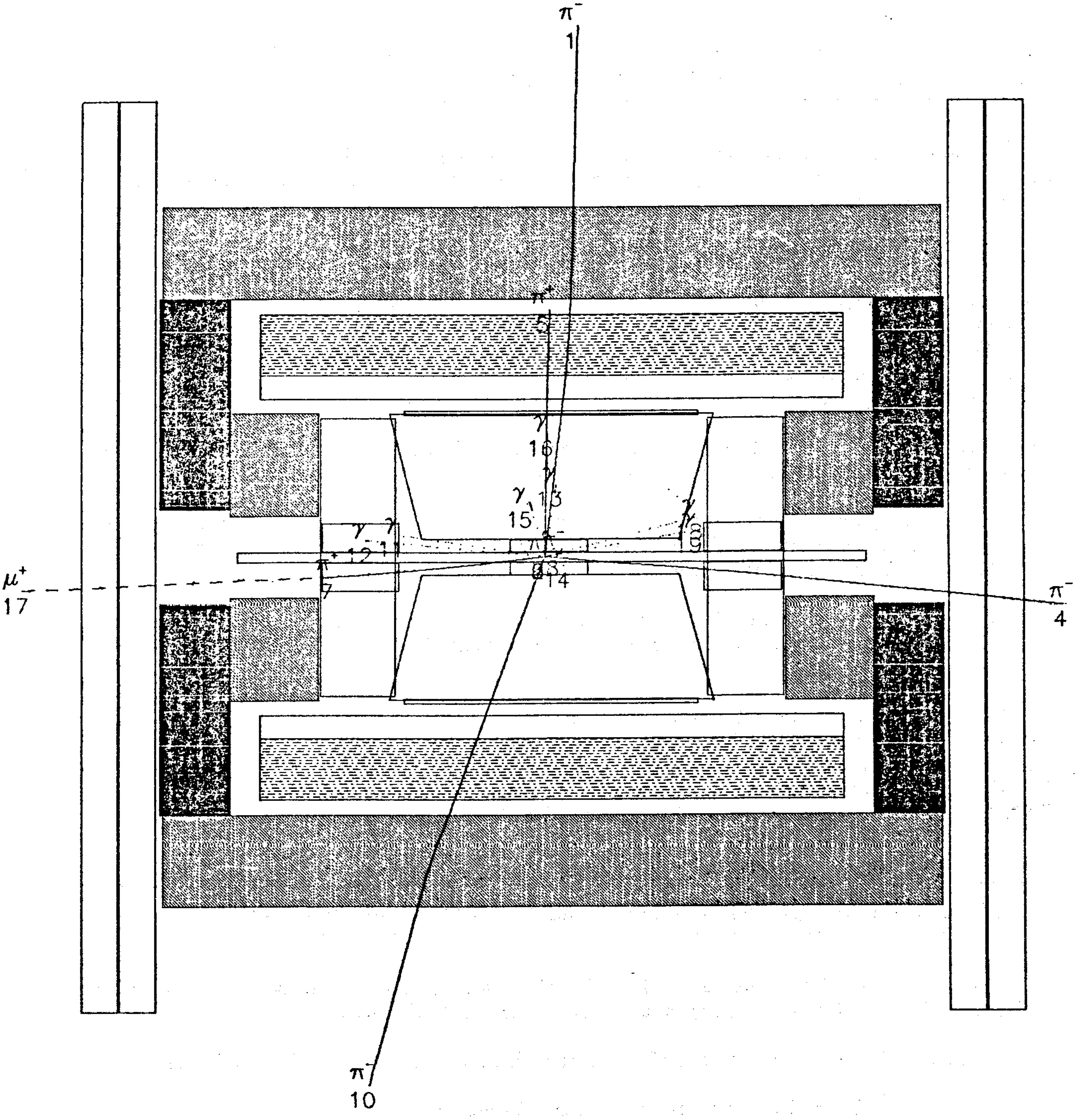
Draws the particle names and/or the track numbers of track ITRA, supposing that its space points had been stored in banks JXYZ. At present only primary tracks are scanned by GDPART and their name/number is placed at the end the track trajectory. The view parameters are taken from /GCDRAW/.

ITRA track number (if 0 all tracks are taken)
ISEL isel=x1 draws the track number, isel=1x draws the particle name, isel=11 draws both
SIZE character size in cm

CALL GDCXYZ

If GDCXYZ is called at tracking time (for instance by GUSTEP), it draws the tracks 'online', i.e. exactly at the same time they are generated by the tracking package of GEANT3, giving so a useful interactive debugging tool. The line style is the same as for GDXYZ. The view parameters are taken from /GCDRAW/.

```
CALL GSATT('HB  ','FILL',3)
CALL GSATT('HE  ','FILL',1)
- - - - -
- - - - -

CALL GDRAWC('OPAL',2,5.,10.,10.,0.013,0.013)
CALL GDXYZ(0)
CALL GDPART(0,11,0.25)
```



Author(s) : R.Brun, W.Gebel†, P.Zanarini
Origin : Same

Submitted: 07.03.84
Revised: 30.05.86

Drawing Track Hits in Sensitive Detectors

```
CALL GDHITS(IUSET,IUDET,ITRA,ISYMB,SSYMB)
```

Draws the hit points as stored by GSAHIT, which were generated by track ITRA in detector IUDET of set IUSET, with the currently selected view parameters in /GCDRAW/. The character plotted at each point may be chosen by ISYMB :

−1	(small) hardware points	(fast)
0	software crosses	(default)
840/850	empty/full circles	(slow)
841/851	empty/full squares	(slow)
842/852	empty/full triangles (up)	(slow)
843/853	empty diamond/full triangle (down)	(slow)
844/854	empty/full stars	(slow)

The numbers correspond to the table on page 25 of the HPLOT manual; any other number given will result in option 0. Except for ISYMB = −1 the size of the character can be chosen by SSYMB.

On the 2D projection on the screen one can distinguish which set/detector a given track passes, by drawing different symbols for the hits in different sets/detectors. The size of these symbols may then be chosen to fit into the scale of the total picture (detectors and tracks).

Note: It is obligatory for the use of this routine that the spatial MARS (MAster Reference System) current coordinates of the hits are stored as the first 3 elements of the hit [HITS 200].

IUSET	user set identifier (if 0 all sets are taken)
IUDET	user detector identifier (if 0 all detectors are taken)
ITRA	number of selected track (if 0 all tracks are taken)
ISYMB	character selection number (see table above)
SSYMB	size of characters in cm (if 0, a default of 0.1 is taken)

```
CALL GDCHIT(IUSET,IUDET,ITRA,ISYMB,SIZMAX,IHIT,HITMIN,HITMAX)
```

Draws the hit points as stored by GSCHIT, which were generated by track ITRA in detector IUDET of set IUSET, with the currently selected view parameters in /GCDRAW/.

Except for ISYMB = −1 the size of the character is chosen as a function of HITS(IHIT) :

$$SIZE = sizmax * (hits(ihit) - hitmin) / hitmax$$

Note: It is obligatory for the use of this routine that the spatial MARS (MAster Reference System) current coordinates of the hits are stored as the first 3 elements of the hit [HITS 200].

SIZMAX	maximum character size in cm
IHIT	HITS(IHIT) contains the calorimeter quantity
HITMIN	lower boundary of HITS(IHIT)
HITMAX	upper boundary of HITS(IHIT)

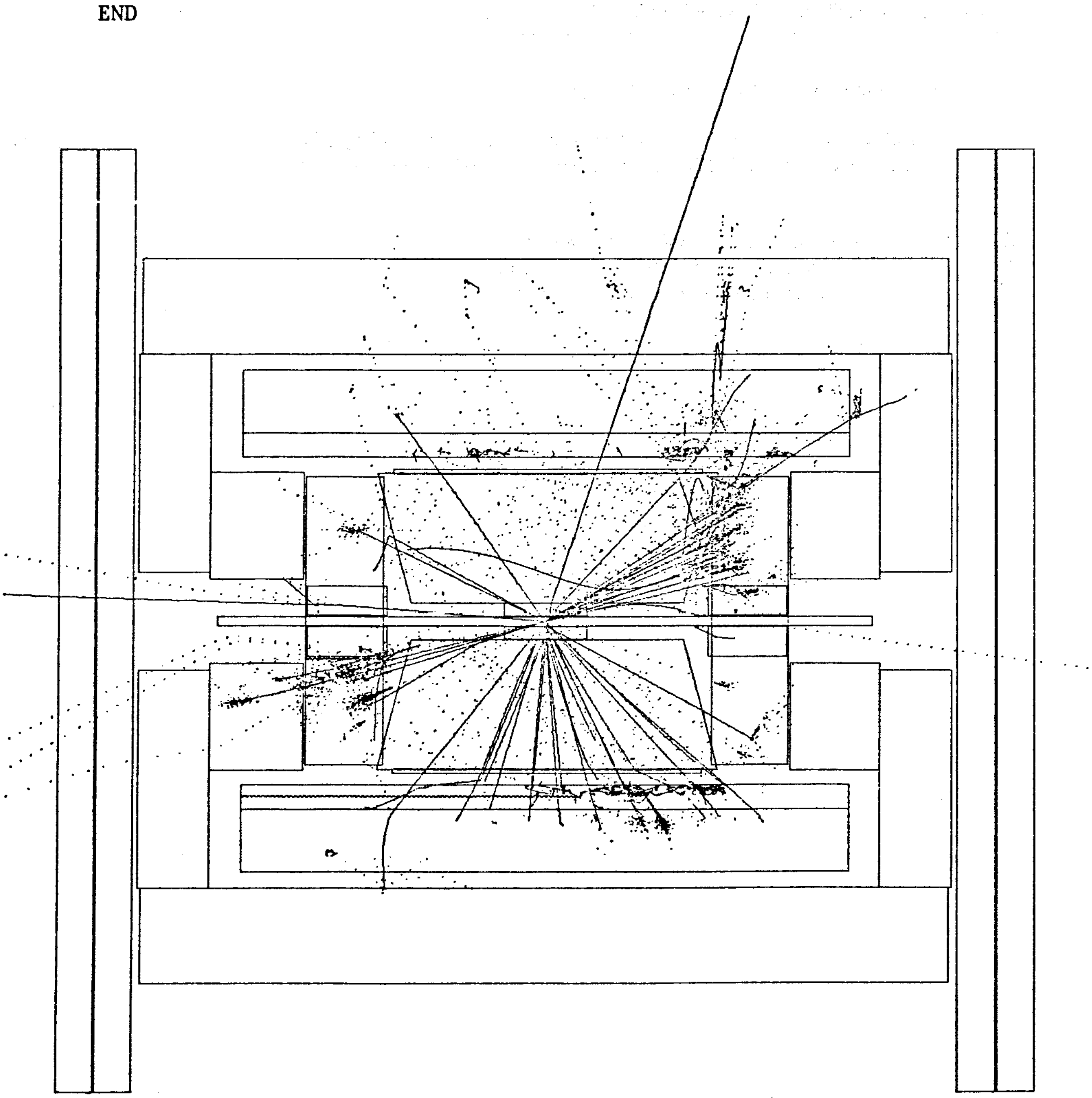
```
CALL GDAHIT(X,Y,Z,ISYMB,SSYMB)
```

Draws one hit point at coordinates X,Y,Z.

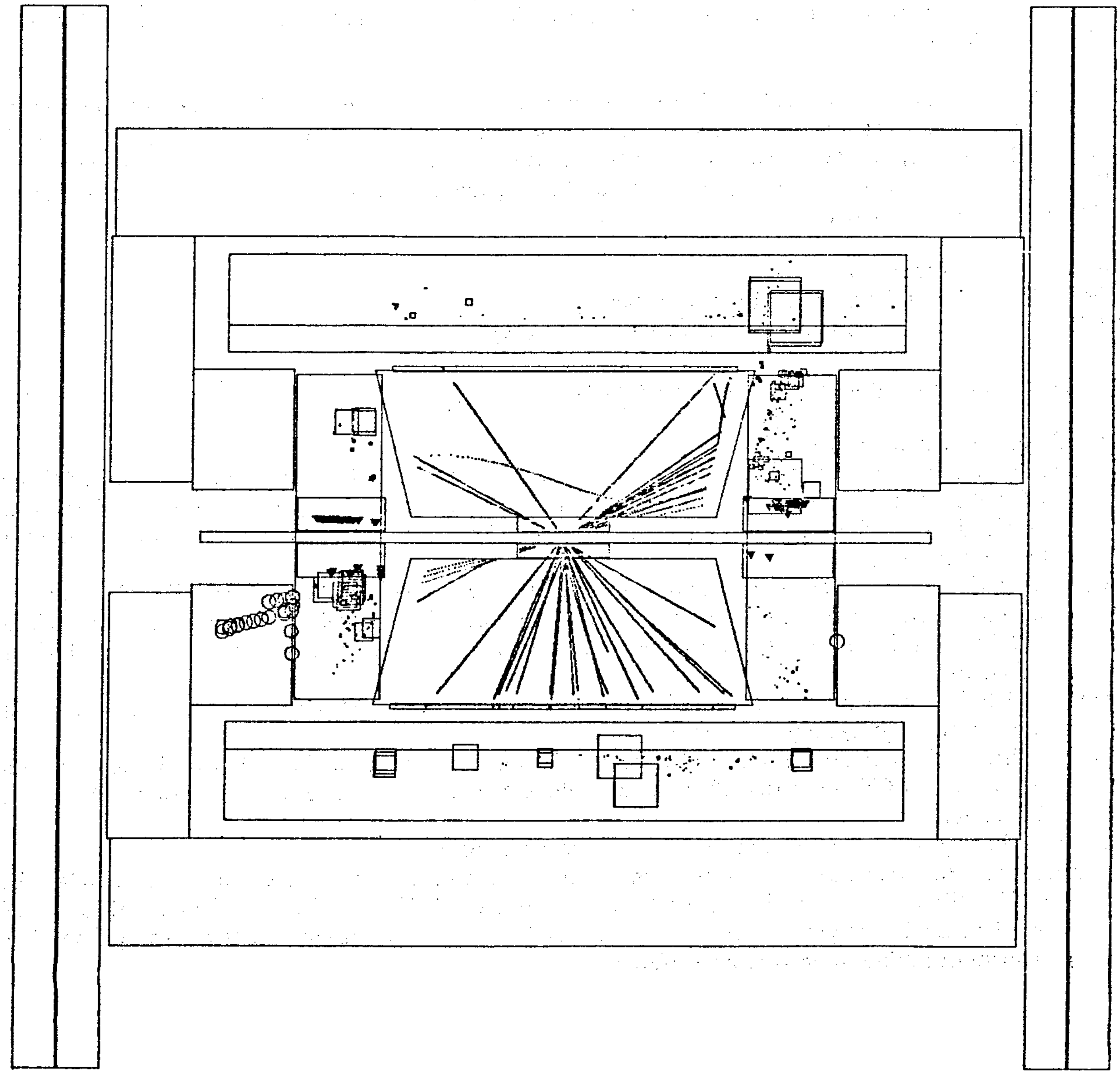
A call to GDAHIT in the user hit routines, gives the possibility of not storing the hits coordinates in the argument vector of GSAHIT, as requested for GDHITS and GDCHIT.

X	x coordinate in MARS of the hit point
Y	y coordinate in MARS of the hit point
Z	z coordinate in MARS of the hit point

```
SUBROUTINE GUTREV
- - - - -
CALL GDOPT('THRZ','ON  ')
CALL GDRAWC('OPAL',2,5.,10.,10.,0.013,0.013)
CALL GTREVE
END

SUBROUTINE GUSTEP
- - - - -
CALL GDCXYZ
- - - - -
END
```



```
CALL GDOPT('THRZ','ON  ')
CALL GDRAWC('OPAL',2,5.,10.,10.,0.013,0.013)
CALL GDCHIT('EB  ',0,0,841,1.,6,0.,0.5)
CALL GDCHIT('EE  ',0,0,841,1.,6,0.,0.5)
CALL GDCHIT('HB  ',0,0,851,1.,5,0.,0.5)
CALL GDCHIT('HE  ',0,0,851,1.,5,0.,0.5)
CALL GDHITS('CJ  ',0,0,0,0.01)
CALL GDHITS('CV  ',0,0,0,0.01)
CALL GDHITS('CZ  ',0,0,0,0.01)
CALL GDHITS('F   ',0,0,853,0.1)
CALL GDHITS('HP  ',0,0,840,0.2)
CALL GDHITS('ME  ',0,0,842,0.2)
CALL GDHITS('MB  ',0,0,842,0.2)
```



Author(s) : P.Zanarini
Origin : Same

Submitted: 15.01.84
Revised: 28.04.86

Drawing the Geometrical Tree

```
CALL GDTREE(NAME,LEVMAX,ISEL)
```

Draws the tree of the volume data structure starting from the volume NAME, for LEVMAX depth levels; a continuation link line is drawn for those elements being at the last level on the picture but having descendents.

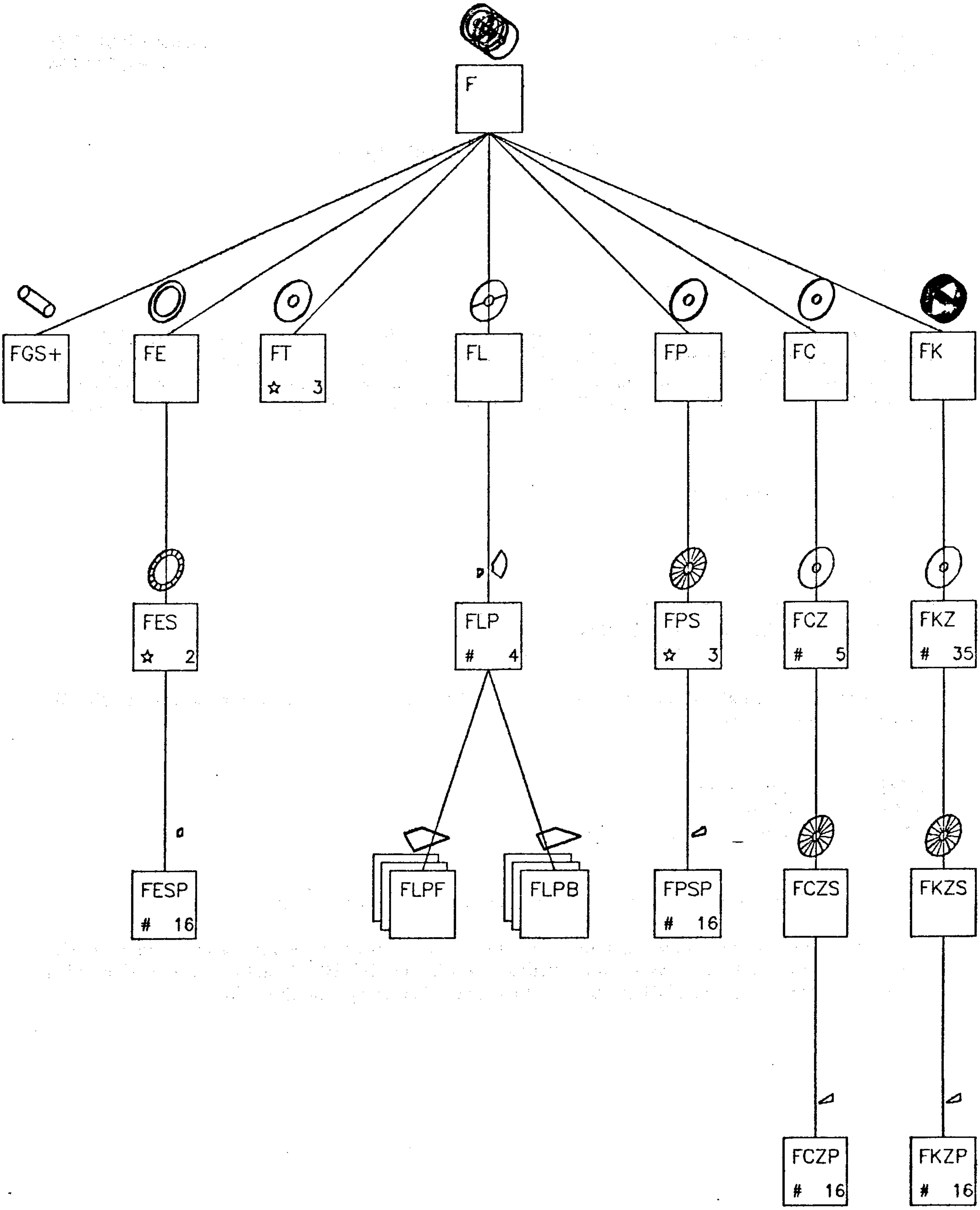
The size of the picture and the dimensions of each element are computed automatically. However, in case of very large trees (i.e. with many different names descending from the same parent), the GSATT routine [GEOM 500] can be called with 'SEEN' atribute −3 for whichever of those different volumes. GDTREE will apply then a special optimization in drawing the tree: only the first, the last, and a generic middle element will be taken.

NAME	volume name
LEVMAX	depth levels (if 0 all levels are taken)
ISEL	options selected (decimal integer, see below)

ISEL is used to select options in the picture of the tree:

if 0, only the node name is drawn
if xxxxx1, multiplicity is added; each node that has been positioned several times (by GSPOS or GSPOSP) has a multiplicity number preceded by the symbol '*'; each node that has been obtained by division mechanism (by GSDVN, GSDVT, etc.) has a multiplicity number preceded by the symbol '#';
if xxxx1x, 'ONLY' information is added; volumes that are NOT-ONLY (so called MANY) will be drawn as three square nodes, one overlapping the others but slightly shifted; (note: if NAME itself is NOT-ONLY it will not recognised as such, because that information is in its mother which is unknown);
if xxx1xx, 'DET ' information is added; if a volumes is a detector it is drawn as a square node with another one slightly larger around it;
if xx1xxx, 'SEEN' information is added; if a volume is not seen (either for its SEEN attribute or for its ascendents SEEN attributes affecting the whole tree path) its square node is drawn by using dashed lines;
if x1xxxx, a little picture of the volume is added above each node;
if 1xxxxx, after the tree has been drawn the graphics cursor is called and returns in IPICK (in /GCTREE/) the node picked; the interactive command DTREE [XINT 110] makes then use of that information to perform various operations, for instance re-drawing the tree starting from that element or setting some volume attributes.

Example of GDTREE output



Author(s) : P.Zanarini
Origin : Same

Submitted: 01.11.83
Revised: 28.04.86

Drawing Volume Specifications

CALL GDSPEC(NAME)

Draws a picture showing all specifications for a given volume. They are: a space view of the volume (from theta=45 and phi=135), a front view cut, a side view cut, the axes, the man, the scale, the shape type and all the numerical parameters that define the volume. In drawing the volumes GDSPEC sets visibility ('SEEN' attribute of GSATT) only for the volume NAME itself and its next level descendents.

NAME	volume name

CALL GDFSPC(NAME,ISORT,INTER)

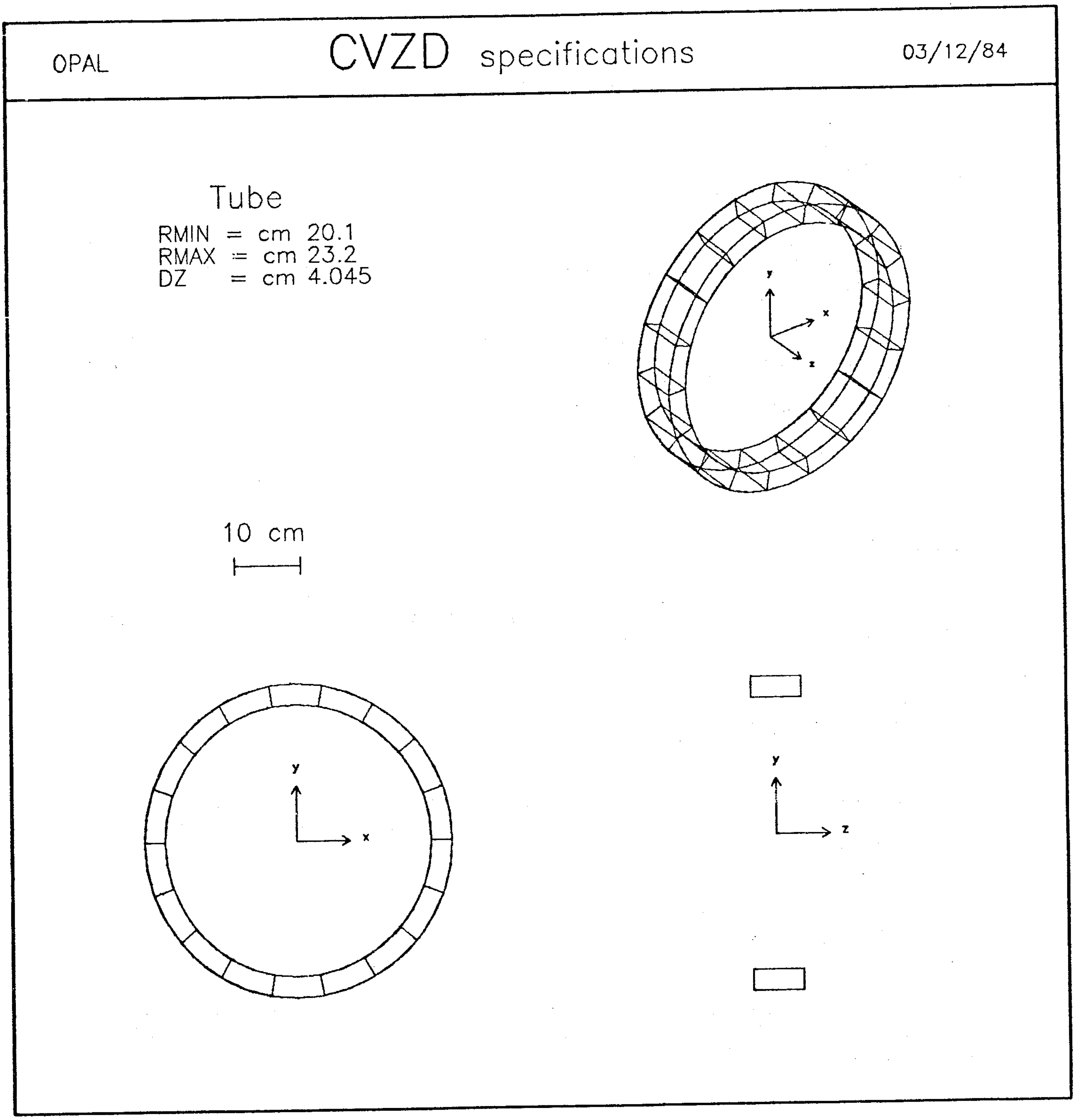
Draws on separate pictures the full set of GDSPEC, i.e. calls GDSPEC for the volume NAME and for all its descendants.

NAME	volume name
ISORT	alphabetic sorting flag
INTER	interactive/batch version flag

If ISORT=1 all the pictures will be drawn in ascending alphabetic order.

If INTER=1 the routine will prompt the user at each plot before doing a clear screen, otherwise it will clear automatically the screen before starting a new frame. INTER should be set to 1 when using the interactive version of GEANT3 and to any other value when using a batch version.

Example of GDSPEC output



OPAL

CVZD specifications

03/12/84

Tube

RMIN = cm 20.1
RMAX = cm 23.2
DZ = cm 4.045

10 cm

Author(s) : P.Zanarini
Origin : R.Brun, P.Zanarini

Submitted: 01.10.83
Revised: 28.04.86

Handling View banks

CALL GDOPEN(IVIEW)

Opens a 'view bank', used to store 2D graphics information coming from the interpretation of 3D structures (but also 2D annotation). All the calls to the drawing routines will fill the view bank IVIEW, without displaying anything on the terminal screen.

For geometrical structures with many different volume names, it can happen that the interpretation of the JVOLUM bank takes much longer than the actual drawing, so it would be worthwhile interpreting JVOLUM only once and saving somewhere the interpreted 2D stream of polylines and text, then drawing the picture as many times as wanted just replaying the 2D information.

IVIEW view number (IVIEW > 0)

CALL GDCLOS

Closes the current view bank opened. Once a view bank has been closed no more drawing can be added to it. A call to GDCLOS also restores the screen-mode.

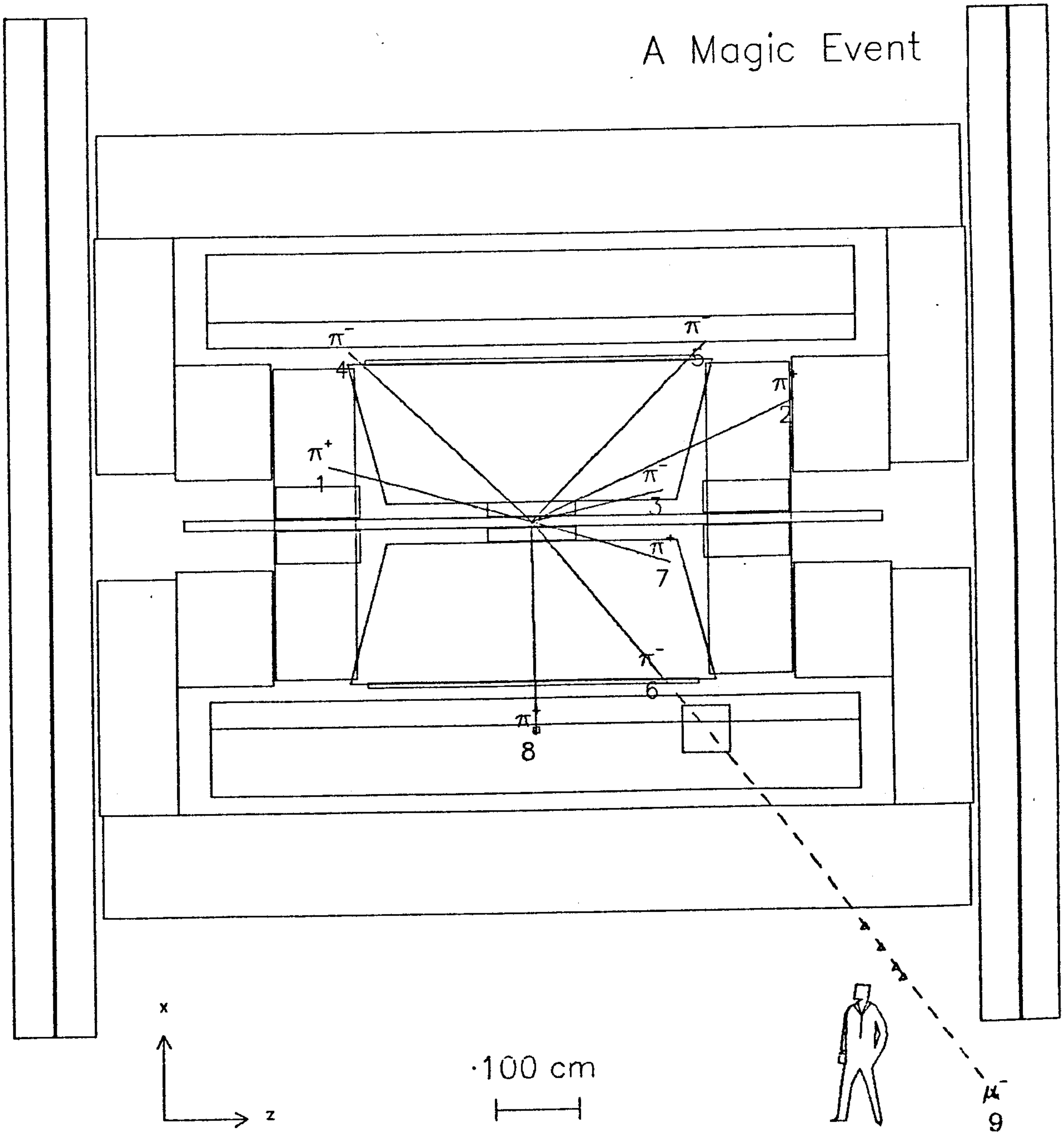
CALL GDSHOW(IVIEW)

Shows a given view. GDSHOW can be called either before or after a view bank has been closed. A bank-to-bank copy from 'isource' to 'idest' is possible by calling GDSHOW(isource) between GDOPEN(idest) and GDCLOS.

CALL GDELET(IVIEW)

Deletes a view bank from memory. If called before a bank has been closed, GDELET will also restore the screen-mode. The view banks protected against delete, i.e. with Q(JDRAW+IVIEW)=3, clearly cannot be deleted by GDELET.

```
CALL GDOPEN(3)
CALL GDRAWC('OPAL',2,5.,10.,10.,0.013,0.013)
CALL GDRAWT(14.,17.,'A M<AGIC> E<VENT$',0.4,0.,1,0)
CALL GDHITS('MB  ',0,0,842,0.1)
CALL GDCHIT('EB  ',0,0,841,1.,6,0.,0.5)
CALL GDXYZ(0)
CALL GDPART(0,11,0.25)
CALL GDMAN(15.,2.)
CALL GDSCAL(10.,1.)
CALL GDAXIS(-700.,0.,-450.,100.)
CALL GDCLOS


CALL GDSHOW(3)
```



Author(s) : P.Zanarini
Origin : R.Brun, P.Zanarini

Submitted: 15.11.83
Revised: 28.04.86

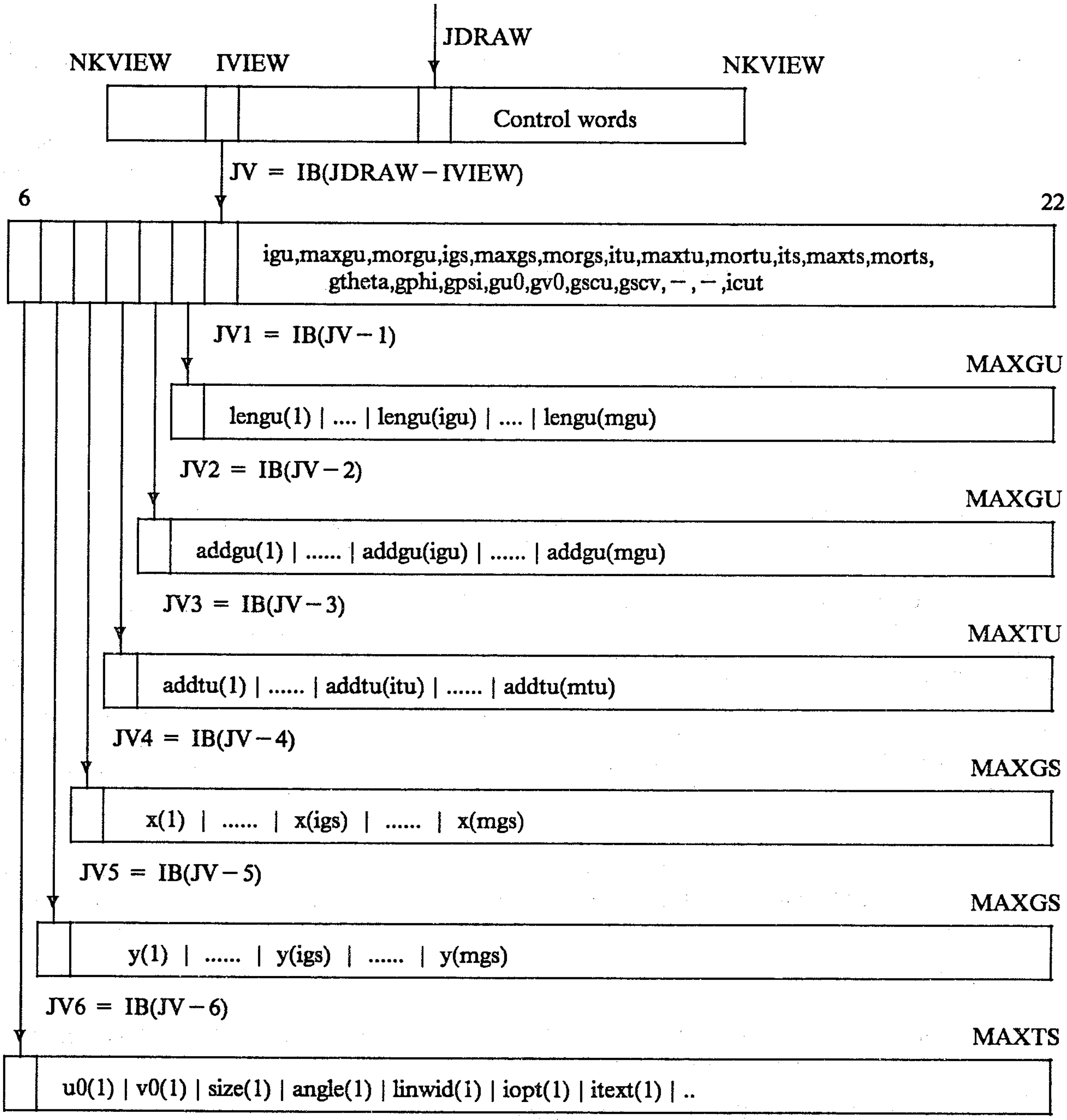
The View data structure JDRAW

NKVIEW	Number of views
IVIEW	Current view selected
IGU	Current graphic unit pointer
MAXGU	Number of graphic units
MORGU	Number of words to push in graphic unit bank
IGS	Current graphic segment pointer
MAXGS	Number of graphic segments
MORGS	Number of words to push in graphic segment bank
ITU	Current text unit pointer
MAXTU	Number of text units
MORTU	Number of words to push in text unit bank
ITS	Current text segment pointer
MAXTS	Number of text segments
MORTS	Number of words to push in text segment bank
LENGU	Array containing: lengths for each graphic unit + LINATT (line attributes)
ADDGU	Array containing addresses for each graphic unit
ADDTU	Array containing addresses for each text unit
X	Array containing u-coordinates of graphic segments
Y	Array containing v-coordinates of graphic segments
ICUT	Cut axis (1,2,3 or 0 if no cut) of the view
LINWID	Text line width + ITXATT (text attributes)

GTHETA, GPHI, GPSI, GU0, GV0, GSCU, GSCV, are the viewing parameters stored in /GCDRAW/. U0, V0, SIZE, ANGLE, IOPT, ITEXT have the same meaning of those given as arguments to GDRAWT (or HPLSOF).

A control word is stored in Q(JDRAW+IVIEW), dividing the view banks in three classes:

=1. for empty banks (created by internal routines to avoid gaps) or for deleted banks
=2. for all the banks created by the user
=3. for protected banks (reserved for internal use); they cannot be deleted by the user


JDRAW
NKVIEW
IVIEW
NKVIEW
Control words
JV = IB(JDRAW – IVIEW)
6
22
igu,maxgu,morgu,igs,maxgs,morgs,itu,maxtu,mortu,its,maxts,morts,
gtheta,gphi,gpsi,gu0,gv0,gscu,gscv, – , – ,icut
JV1 = IB(JV – 1)
MAXGU
lengu(1) | .... | lengu(igu) | .... | lengu(mgu)
JV2 = IB(JV – 2)
MAXGU
addgu(1) | ...... | addgu(igu) | ...... | addgu(mgu)
JV3 = IB(JV – 3)
MAXTU
addtu(1) | ...... | addtu(itu) | ...... | addtu(mtu)
JV4 = IB(JV – 4)
MAXGS
x(1) | ...... | x(igs) | ...... | x(mgs)
JV5 = IB(JV – 5)
MAXGS
y(1) | ...... | y(igs) | ...... | y(mgs)
JV6 = IB(JV – 6)
MAXTS
u0(1) | v0(1) | size(1) | angle(1) | linwid(1) | iopt(1) | itext(1) | ..

Author(s) : P.Zanarini
Origin : Same

Submitted: 15.08.83
Revised: 28.04.86

Other routines of the Drawing package

CALL GDOPT(IOPT,IVAL)

Sets drawing options.

The option 'PROJ' selects the type of projection, and can have either the value 'PARA' for orthographic parallel projection (the default) or 'PERS' for perspective projection.

The option 'THRZ' is the acronym of Tracks and Hits in a Radius versus Z projection, where R is the vertical axis and Z the horizontal one. This special projection, if activated, is actually used only in drawing tracks and hits, and only if currently viewing an X-Z cut (cutthe=90 and cutphi=90 in GDRAWX, icut=2 in GDRAWC) or a Y-Z cut (cutthe=90 and cutphi=180 in GDRAWX, icut=1 in GDRAWC) of the detector. The option can have the value '360 ' for a rotation around 360 degrees, so all the projected points will have R positive; '180 ' or 'ON ' for a rotation around 180 degrees, so the points will have R positive or negative depending on their Y coordinate in the Projection Reference System; 'OFF ' (the default) for not activating the R-Z projection and having the standard X-Z or Y-Z projections.

Note: When 'THRZ' option is set to 'ON ' or '180 ' an effect of particle reflection will be noticed for particles originated in one half space and going to the other half: when crossing the horizontal axis they will be reflected back on the original half space. If this way of drawing would cause problems to the user, the option '360 ' can be chosen, by which all the tracks (and hits) are on the top (positive) half space.

The option 'TRAK' selects wether the lines joining track points are drawn by GDXYZ ('LINE', the default) or if only the track points are drawn ('POIN').

IOPT option name : 'PROJ'/'THRZ'/'TRAK'
IVAL option value :
'PROJ' = 'PARA'/'PERS'
'THRZ' = 'ON '/'OFF '/'180 '/'360 '
'TRAK' = 'LINE'/'POIN'

CALL GDZOOM(ZFU,ZFV,UZ0,VZ0,U0,V0)

This routine sets the zoom parameters (part of the viewing parameters in /GCDRAW/) that define how the various objects (detector, tracks, hits, etc.) will be displayed by next drawing operations.

The zoom is computed around UZ0,VZ0 (user coordinates). This point will be moved to U0,V0 in the resulting picture. ZFU,ZFV are the zoom factors (positive). If ZFU or ZFV are zero, the zoom parameters are reset, and the original viewing is restored.

ZFU	zoom factor for u-coordinate
ZFV	zoom factor for v-coordinate
UZ0	u-coordinate of centre of zoom
VZ0	v-coordinate of centre of zoom
U0	u-coordinate of centre of zoom in the resulting picture
V0	v-coordinate of centre of zoom in the resulting picture

```
CALL GDAXIS(X0,Y0,Z0,AXSIZ)
```

Draws the axes of the MAster Reference System, corresponding to the current viewing parameters. All the arguments are in MARS units.

X0	x-coordinate of origin
Y0	y-coordinate of origin
Z0	z-coordinate of origin
AXSIZ	axis size

```
CALL GDSCAL(U0,V0)
```

Draws a scale corresponding to the current viewing parameters. Seven kinds of units are available, from 1 micron to 100 cm. The best unit is automatically selected.

U0	u-coordinate of the centre of the scale
V0	v-coordinate of the centre of the scale

```
CALL GDMAN(U0,V0)
```

Draws the profile of a man, in 2D user coordinates, within current scale factors. The man is approximately 160 cm high.

U0	u-coordinate of the centre of the man
V0	v-coordinate of the centre of the man

```
CALL GDRAWT(U,V,ITEXT,SIZE,ANGLE,LWIDTH,IOPT)
```

Draws text with software characters. It has the same arguments of the HPLOT routine HPLSOF (a call to HPLSOF is actually performed).

U	u-position of text string
V	v-position of text string
ITEXT	text (string terminated by a dollar sign)
SIZE	character size (cm)
ANGLE	rotation angle (degrees)
LWIDTH	line width (1,2,3,4,5)
IOPT	centering option for the text

-1 = left adjusted
0 = centered
1 = right adjusted

CALL GDRAWV(U,V,NP)

Draws 2D polylines in user coordinates. The routine GDFR3D can be called to transform 3D points in 2D user coordinates.

U array of u-coordinates
V array of v-coordinates
NP number of vectors

CALL GDHEAD(ISEL,NAME,CHRSIZ)

Draws a frame header. Different options can be chosen by the argument ISEL.

ISEL option to be selected for the title name (decimal integer):
ISEL=0 to have only the header lines
ISEL=xxxxx1 to add the text NAME centered on top of header
ISEL=xxxx1x to add global detector name (first volume) on left
ISEL=xxx1xx to add date on right
ISEL=xx1xxx to select thick characters for text in top of header (i.e. with larger line width)
ISEL=x1xxxx to add the text 'EVENT NR x' on top of header
ISEL=1xxxxx to add the text 'RUN NR x' on top of header
Note that ISEL=x1xxx1 or ISEL=1xxxx1 are illegal choices, i.e. they generate overwritten text.
NAME title (string terminated by a dollar sign)
CHRSIZ character size (cm) of text NAME

CALL GDCOL(ICOL)

Sets the colour code.

ICOL can be positive, negative, or zero.
ICOL $<$ 0 to set the colour permanently;
ICOL $>$ 0 to set the colour temporarily;
ICOL = 0 to restore the permanent colour value.

For example we can set ICOL= -1 to start with colour # 1, then change to ICOL=3 to set colour # 3, and at the end restore the original colour by setting ICOL=0 that takes colour # 1 again.

ICOL colour code (1,2,...,8)

```
CALL GDLW(LW)
```

Sets the line width.
LW can be positive, negative, or zero.
LW < 0 to set the line width permanently;
LW > 0 to set the line width temporarily;
LW = 0 to restore the permanent line width value.

For example we can set LW = −1 to start with line width # 1, then change to LW = 3 to set line width # 3, and at the end restore the original line width by setting LW = 0 that takes line width # 1 again.

LW	line width code (1,2,...,5)

```
CALL GDCURS(U0*,V0*,ICHAR*)
```

If this routine is called in interactive version of GEANT3 and with PIGS or GKS option set, it draws on the screen the present position of the graphics cursor, then moving UP-DOWN and LEFT-RIGHT arrows (or equivalent) the user goes on the screen point he wants be selected, and pressing a key (but not the carriage return) that point is held. The routine returns then in U0,V0 the user coordinates of the graphics cursor, and in ICHAR the ASCII value of the key pressed.
If the user types the carriage return, the previous value is retained.

U0	u-coordinate of the graphics cursor
V0	v-coordinate of the graphics cursor
ICHAR	ASCII value of the key pressed

```
CALL GDFR3D(X,NPOINT,U*,V*)
```

Converts from 3D space coordinates (either in MAster or Daughter Reference System) to the corresponding 2D user coordinates.

X	array X(3,NPOINT) of space points
NPOINT	number of points: If NPOINT > 0 then X is in the current DRS If NPOINT < 0 then X is in MARS
U	array of NPOINT u-coordinates
V	array of NPOINT v-coordinates

This routine maps a space point (X,Y,Z on a right-handed reference system) onto a plane perpendicular to the observer's line of sight, defined by the spherical angles theta,phi:

THETA, the angle between the line of sight and the Z-axis. (from 0. to 180. degrees)
PHI, the angle between the X-axis and the projection onto the X-Y plane of the line of sight (from 0. to 360. degrees)

The plane onto which the point is mapped is actually the X-Y plane of the Projection Reference System, and the observer's line of sight is the Z-axis of PRS. The vertical axis on this plane (Y-axis on PRS) is obtained by intersecting this plane with the one of MARS or DRS containing the line of sight and the Y-axis.

```
CALL GD3D3D(XIN,NPOINT,XOUT*)
```

Converts from 3D space coordinates in MARS to 3D coordinates in Projection Reference System (whose z-axis is along the line of sight given by theta and phi angles).

XIN	array XIN(3,NPOINT) of input points in MARS
NPOINT	number of points.
XOUT	array XOUT(3,NPOINT) of output points in PRS

Author(s) : P.Zanarini
Origin : Same

Submitted: 25.05.85
Revised: 28.04.86

The GEANT3/PIONS Interface

An interface of GEANT3 to PIONS[1] is available in patch GPIONS of GEANX pam. While PIONS can be run on a wide range of terminals, the main purpose of this interface is to provide dynamic graphics on terminals with 3D in hardware (like the Megatek). The interface supports currently three kinds of PIONS 'workstations' : Megatek, Apollo, and raster terminals.

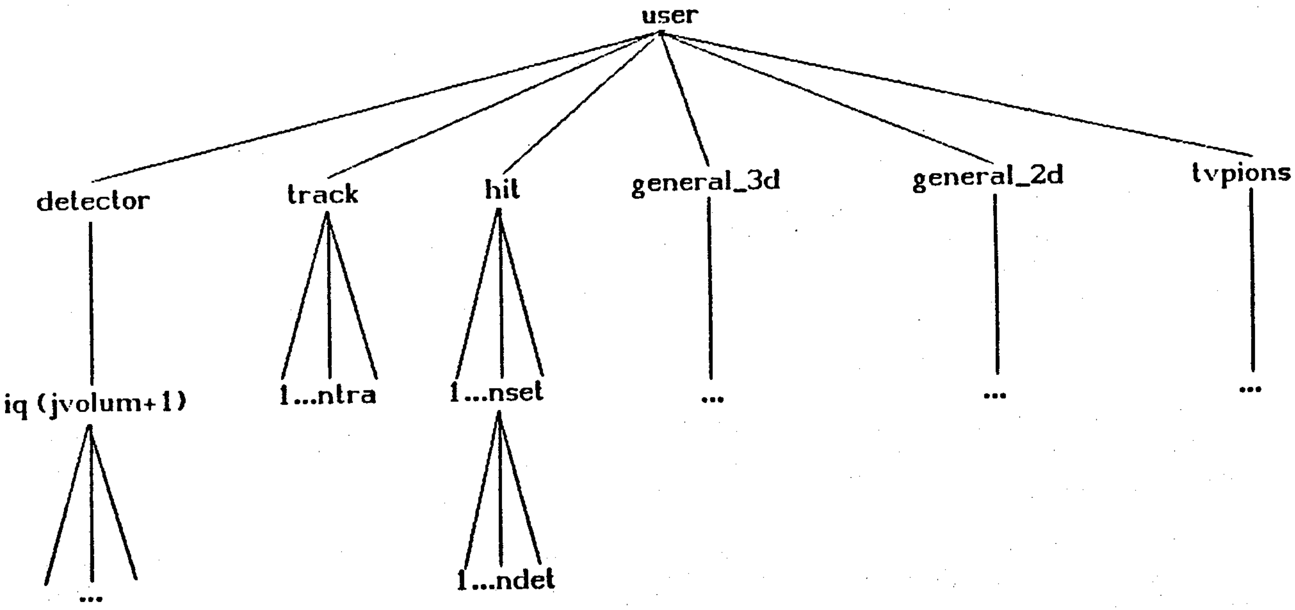
To activate the PIONS interface the GPIONS object module has to be loaded together with GEANT3. Moreover, the usual interactive environment of GEANT3, with the command processor ZCEDEX,[2] can be used as well. In the same interactive program the user can actually switch from GEANT3 to PIONS and viceversa.

When GPIONS is loaded, GEANT will not have any 2D graphics (eg. GDTREE, GDRAWT, GDMAN, etc.) : only 3D objects will be shown by the PIONS package.

Three interactive commands [XINT 120] have been provided to use PIONS from GEANT3 :

- PIONS-OPEN to start the filling of the PIONS graphics structure
- PIONS-CLOSE to finish the filling of the PIONS graphics structure
- PIONS-GO to give the control to PIONS

The drawings are structured inside PIONS in the following way :



[1] The PIONS User Guide - CERN/DD/EE/85-1
PIONS, A High Performance Graphics Package - CERN/DD/EE/85-2

[2] ZCEDEX User Guide - CERN/DD/EE/80-6

The graphics menu of PIONS is set up in the GPIONS routine of the GPIONS package. The commands currently implemented can be subdivided in the following classes.

Viewing operations (the parameters are entered from the keyboard):

REFP to set the view reference point (i.e. where the line of sight intersects the projection plane, initialized to 0.,0.,0.);
the term 'view reference point' must not be confused with the 'view point', the latter being used anywhere in the GEANT3 documentation and corresponding to the 'projection direction/point' of PIONS (see below)

PROJ to set the projection direction/point (i.e. where the observer's eye stands, initialized to -1.,1.,1.); used to move/rotate the picture on the screen

2D VWPT to set the 2D viewport boundary (initialized to +5 ... +95 for horizontal and vertical axes; the full screen goes from 0 to 100, in normalized device coordinates)

3D WNDW to set the 3D window boundary (initialized to -1000 ... +1000 for the three axes of world coordinates); used to scale the picture on the screen;
note that PIONS world coordinates correspond to MARS coordinates in GEANT3

Pick range definitions (no input required):

ALL to select the path on the PIONS tree at the level /user

DETECTOR to select the path on the PIONS tree at the level /user/detector

TRACK to select the path on the PIONS tree at the level /user/track

HIT to select the path on the PIONS tree at the level /user/hit

Pick operations:

LOCATE to locate something (i.e. to get the world coordinates of the point being input through the locator)

PICK to pick something (i.e. to identify the object whose point being input through the pick belongs)

HIDE to hide something (i.e. to switch off the visibility flag of the object whose point being input through the pick belongs)

DROP to drop all the objects from a given level in the tree (no input required)

Others operations (no input required, except for 'menu'):

CAPTURE to produce a capture file (the command is activated only if the capture was requested from the CLINIT procedure, initializing the PIONS client)

ECHO ON to switch on the pick/locator feedback on the graphics terminal

ECHO OFF to switch off the pick/locator feedback on the graphics terminal

REFR to refresh the screen

MENU to move on the screen the graphics menu (use the locator to input the new top left corner of menu)

GEANT3 to return to GEANT3

The user may control the volumes that have to go in the PIONS structure, by the 'SEEN' attribute of GSATT (GEOM 500). In other words, what one would see on a 2D terminal without using PIONS, that will be pushed in the PIONS tree structure.

The 'NODE' attribute has to be used also; for each volume having this attribute, a PIONS directory will be created (when drawing after the PIONS-OPEN command). It is recommended to set always 'NODE' attribute for the top volume of the geometrical description.

Author(s) : P.Zanarini
Origin : Same

Submitted: 01.08.85
Revised: 14.05.86

The GEANT3/Apollo-GMR Interface

An interface of GEANT3 to the Apollo Graphics Metafile Resource GMR-3D is available in patch GGMR of GEANX pam. The main purpose of this interface is to take advantage of the speed of GMR in real-time manipulation (zooming, moving and rotating) of the picture on the Apollo.

To activate the GMR interface the GGMR object module has to be loaded together with GEANT3. Moreover, the usual interactive environment of GEANT3, with the command processor ZCEDEX,[1] can be used as well. In the same interactive program the user can actually switch from GEANT3 to GMR and viceversa.

When GGMR is loaded, GEANT will not have any 2D graphics (eg. GDTREE, GDRAWT, GDMAN, etc.) : only 3D objects will be shown by the GMR package.
Two interactive commands [XINT 120] have been provided to use GMR from GEANT3 :

- GMR-OPEN to start the filling of the GMR graphics structure
- GMR-CLOSE to finish the filling of the GMR graphics structure and give the control to GMR

The menu items currently implemented are:

SELECT AXIS to select a view along one axis (then push a mouse button: left for X-view, middle for Y-view, right for Z-view)

ROTATE C.W. to rotate the picture clockwise on the viewing plane by a fixed angle (default = 10 degrees, but can be changed by 'Rot./Zoom mag.' menu)

ROTATE A.C.W. to rotate the picture anti-clockwise on the viewing plane by a fixed angle (default = 10 degrees, but can be changed by 'Rot./Zoom mag.' menu)

ZOOM IN to make the picture bigger by a fixed factor (default = 1.25 times, but can be changed by 'Rot./Zoom mag.' menu)

ZOOM OUT to make the picture smaller by a fixed factor (default = 1.25 times, but can be changed by 'Rot./Zoom mag.' menu)

MOVE to move the picture on the viewing plane (push the middle button of the mouse to select the new centre of the vindow)

RESET VIEW to reset the original parameters for window dimensions and viewing pipepline (i.e. the effect of the first six menu items)

ROT./ZOOM MAG. to change the rotate and zoom factors (push the left button of the mouse to divide the factors by 2, the right button to multiply by 2, the middle button to reset the factors to the original value of 1)

EXIT to return to GEANT3

The user may control the volumes that have to go in the GMR structure, by the 'SEEN' attribute of GSATT. In other words, what one would see on a 2D terminal without using GMR, that will be pushed in the GMR tree structure.

[1] ZCEDEX User Guide - CERN/DD/EE/80-6

Author(s) : F.Bruyant
Origin : R.Brun, F.Bruyant, A.McPherson

Submitted: 01.10.84
Revised: 28.04.86

Introduction to the Geometry package

1. The geometry package

The geometry package consists of subroutines which can be used in the initialisation phase of the program to describe the geometry of the experimental setup, and of subprograms which ensure the communication with the tracking package during the event processing phase.

The following paragraphs review the concepts of the geometry package and explain how the geometrical information should be provided by the user.

It is important to point out that, once the setup has been initialized, the tracking of particles through the different media can proceed without any other intervention from the user [TRAK]. The connection between the geometry and tracking packages is established by the subprograms GMEDIA, GNEXT and GINVOL which answer respectively the questions:

- In which medium is a given point ?
- What is the path length to the next medium ?
- Is a given point still in the current medium ?

2. The volume definition

Experimental setups, as complex as the detectors prepared for LEP, can be described rather accurately through the definition of a set of simple VOLUMES.

Each VOLUME is given a NAME and is characterized by:

- a SHAPE identifier, specifying one of the basic geometrical shapes available [GEOM010],
- the shape parameters, giving the dimensions of the volume,
- a local reference system, whose origin and axes are the ones defined for the given shape (cartesian, cylindrical or spherical coordinates),
- its physical properties, given by a set of constants for the homogeneous MATERIAL which fills the volume [CONS],
- additional properties, known as 'TRACKING MEDIUM' parameters, which depend on the characteristics of the volume itself (the MATERIAL identifier is one of the constants) and on its geometrical and physical environment (properties of the neighbour media, magnetic field, etc.) [CONS,TRAK],
- a set of attributes, in connection with the drawing package and the detector response package [DRAW,HITS].

As long as it is not 'positioned' in a given reference frame, a VOLUME is an entity which has no spatial relation with the setup. By convention, a unique initial volume has to be defined which should match (or surround) the outside boundaries of the entire setup. The reference frame attached to this volume is considered to be the master reference frame.

3. Volumes with contents

A VOLUME can be declared to have 'contents' and become a 'mother' volume. The contents are ***either*** predefined volumes which are explicitly positioned inside the mother, ***or*** new volumes which are implicitly defined by a division mechanism applied to the mother.

Positioning a volume with given shape and dimensions inside a mother volume is achieved by specifying its translation and rotation with respect to the mother reference frame. The user should make sure that no volume overlaps the boundaries of its mother. When a volume is positioned, the user gives it a NUMBER. Multiple copies of a given volume, can be positioned inside a mother with different numbers, or inside different mothers and the contents of the volume are reproduced implicitly in all copies.

Divisions can be performed along any of the three axes of the mother volume. The definition of the axes (X,Y,Z or R_{xy},ϕ,Z or R,θ,ϕ) depends on the shape. The mother volume can be partially or totally divided. The division generates a 'cell', which is considered as a new volume with (usually) the same shape as the mother. Its dimensions are computed according to the declared division number and/or step size. A cell, as any volume, can again be divided along any of its proper axes, or have other volumes positioned into it. Volumes positioned within a cell are reproduced implicitly in all cells.

These operations permit a physical tree to be defined of volumes at deeper and deeper levels.

It is assumed that the 'tracking medium' properties of the contents replace the ones of the mother within the space region they occupy.

A VOLUME is therefcre defined not only by its intrinsic characteristics but also by the definition of its 'descendants', namely its contents, the contents of its contents, etc.

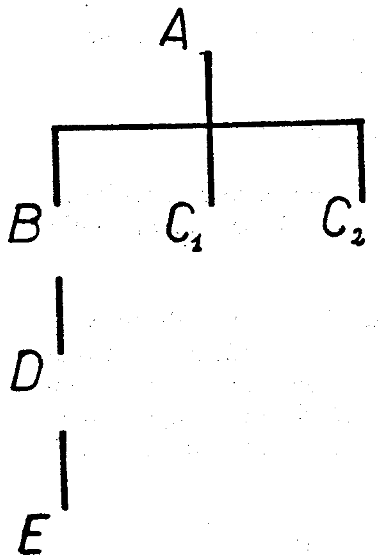
4. Overlapping volumes

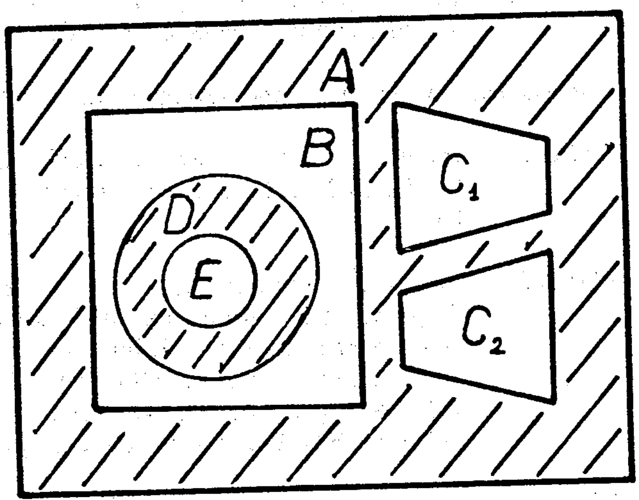
The user may define volumes which have nothing to do with the real physical structure. It is sometimes convenient to make use of such volumes, to artificially delimit regions with simple shapes. As a consequence, it may happen that volumes overlap each other. (A volume positioned inside a mother is obviously not regarded as overlapping the mother).

The handling of overlapping volumes has some implications that the user should be aware of:
A flag 'only/not only' is attached to each positioned volume. The 'not only' option indicates that a point found to be in this volume could also be in other volumes which are not direct descendants of it. The user is free to declare one of the overlapping volumes as 'only', in which case the searching subroutines will give priority to this volume. If a point is inside several 'not only' volumes and outside all 'only' ones, priority will be given to the first volume found at the deepest level and, in order to avoid ambiguities, two overlapping 'not only' volumes should in general be assigned the same default tracking medium.

5. The physical tree

An example of a 4-level physical tree with embedded volumes and the corresponding geometrical configuration are sketched below:





where A and B are ***BOX***es, C a ***TRAP***ezoid, and D and E ***TUBE***s.

Notice that the same physical configuration could be described as well though a 3-level tree if D were defined as a 'hollow' TUBE, with inner radius non zero, and E directly positioned into B.

The package accepts a maximum of 15 levels, which should be enough to represent even the fine details of a complex setup.

6. The data structure JVOLUM and the COMMON block /GCVOLU/

In practice, the physical tree is not represented as such in the data part of the program. Instead, a logical tree structure is defined, the JVOLUM data structure, which describes the arrangement of volumes in a compact and recurrent way. Each generic volume appears once, and once only, and carries the information relevant to the volume itself and to its contents, if any, by reference to the generic volumes corresponding to those contents.

In the situation where division or multiple copies occur, there is no longer a one-to-one correspondence between a given volume in the logical tree and a unique region in space. Information has to be kept at tracking time to identify which division cell or which copy was considered at each depth level along the path through the physical tree. This information is stored by the subroutine GMEDIA, for the current point of the current track, in the common /GCVOLU/. It contains the current level number NLEVEL and, for each level, starting from the first initial reference volume, the identification of the corresponding volume, e.g.: NAME, NUMBER, 'ONLY/NOT ONLY' flag, SHAPE parameters, translation and rotation with respect to the master reference frame.

7. The basic user tools

The rules of the game being established, it is easy to introduce the set of subroutines which ensure the functionality of the package.

The user can define a volume through a call to the subroutine:

GSVOLU: Input arguments specify the NAME, SHAPE and parameters of the volume. An output argument returns the position of the volume inside the bank JVOLUM.

The user can position a volume through a call to either one of the following subroutines:

GSPOS: Input arguments specify the NAME and copy NUMBER of the volume to be positioned, the NAME of the mother, the translation and the rotation matrix, and the 'only/not only' flag.

GSPOSP In case the user has to position, inside a mother, a large number of volumes with the same shape but different dimensions (lead glass blocks, BGO crystals, etc.) an alternative is proposed which consists of defining the generic volume once, with the number of shape parameters set to zero, and to call *GSPOSP* for each volume in turn, with the same arguments as GSPOS plus the shape parameters. The volumes will be identified by their NAME + NUMBER as for the multiple copies.

The user can divide a volume through a call to either one of the following subroutines:

GSDVN: Input arguments specify the NAME of the cell volume, the name of the MOTHER being divided, the number of divisions NDIV and the axis along which the division is performed. In this simple case, the cell tracking medium is assumed to be the same as for the mother. See also GSDVN2.

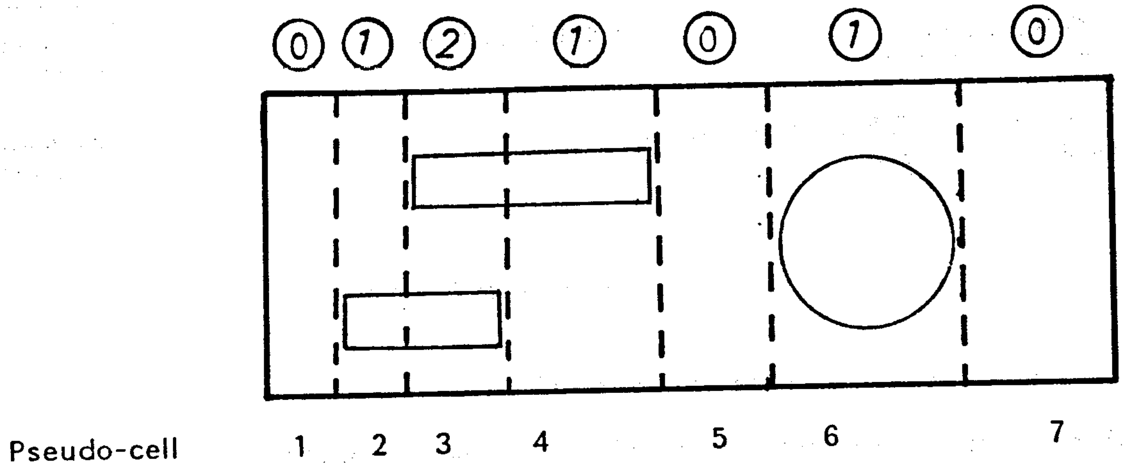
GSDVT: The division STEP is given instead of NDIV, the cell tracking medium is specified and the expected (maximum) number of divisions is given. See also GSDVT2.

GSDVX: In addition to both STEP and NDIV (with at least one of them positive to be effectively useful), the origin of the first cell, the cell tracking medium and eventually the computed (maximum) number of divisions are specified.

8. The optimisation tools

When a track enters a volume with contents, the search time to identify whether the current point is in the mother or in any of the contents is very short when the contents are division cells (straightforward computation along the relevant axis). When the contents have been positioned, the search time can be quite large. In order to save time the user can make use of either of the two following facilities:

1) *GSORD*
From the known position of the contents inside a given volume the subroutine GSORD computes fictitious boundaries along the specified coordinate, simulating a division with non regular step size. A binary search technique is used to identify within which pseudo-cell the current point is. The slow process of computing whether the point is inside or outside the contents is therefore limited to the few (if any) volumes sitting in that pseudo-cell, as sketched below:



The coordinate selected for the pseudo division can be any of X, Y, Z, R_{xy}, R, ϕ or θ. For a given volume the call to GSORD has to come after all its first level contents have been positioned.

2) *GSNEXT*
when a particle ***enters*** a mother volume, the contents are scanned initially in the order they have been positioned, and the user should take care over the best sequence of GSPOS calls. However, when the particle ***comes back*** inside the mother from any one of the contents, it is usually possible to limit the search to the neighbour contents. The subroutine GSNEXT permits the user to inject at initialisation time, for each content in turn, the list of neighbours to search for. A proper use of this facility can reduce the search time significantly.

GSSEAR/GOSEAR
The order in which the contents of a given volume are scanned, as well as the order within the lists of neighbours (if any), can be updated according to the statistics observed during the execution of the program provided the user has explicitly called the subroutine GSSEAR for this volume. The reordering takes place, with control print-out, every time the user calls the subroutine GOSEAR.

9. Geometrical information retrieval

For any volume specified by the description of the path one has to follow through the physical volume tree in order to reach it (namely by the lists of NAMES and NUMBERs at all levels from the top down to the given volume), the geometrical parameters which define its dimensions and position in the MAster Reference System can be loaded in the common /GCVOLU/ by the user through a call to the subroutine GLVOLU.

Author(s) : R.Brun, A.McPherson
Origin : Same

Submitted: 15.08.83
Revised: 22.04.86

The System Shapes

The system shapes supported at present are as follows:

(For each shape the local reference system is shown in the corresponding Figure)

1 'BOX ' is a box. It has 3 parameters, the half lengths in x, y and z.

2 'TRD1' is a trapezoid with only the x length varying with z. It has 4 parameters, the half length in x at the low z surface, that at the high z surface, the half length in y and in z.

3 'TRD2' is a trapezoid with both x and y lengths varying with z. It has 5 parameters, the half length in x at the low z surface, that at the high z surface, the half length in y at the low z surface, that at the high z surface, the half length in z.

4 'TRAP' is a general trapezoid, i.e. one for which the faces perpendicular to z are trapezia and their centres are not at the same x, y. It has 11 parameters: Dz the half length in z, Th & Phi the polar angles from the centre of the face at $z=-Dz$ to that at $z=+Dz$, H1 the half length in y at $z=-Dz$, LB1 the half length in x at $z=-Dz$ and y=low edge, LH1 the half length in x at $z=-Dz$ and y= high edge, Th1 the angle w.r.t. the y axis from the centre of the low y edge to the centre of the high y edge, and H2, LB2, LH2, Th2 the corresponding quantities at $z=+Dz$.

5 'TUBE' is a tube. It has 3 parameters, the inside radius, the outside radius and the half length in z.

6 'TUBS' is a phi segment of a tube. It has 5 parameters, the same 3 as 'TUBE' plus the phi limits. The segment starts at the first limit and includes increasing phi value up to the second limit or that plus 360 degrees.

7 'CONE' is a conical tube. It has 5 parameters, the half length in z, the inside and outside radii at the low z limit and those at the high z limit.

8 'CONS' is a phi segment of a conical tube. It has 7 parameters, the same 5 as 'CONE' plus the phi limits.

9 'SPHE' is a segment of a spherical shell. It has 6 parameters, the inside radius, the outside radius, the theta limits and the phi limits. At present, for the drawing package only first two parameters are significant (inside radius and outside radius) and such a shape is always drawn as a full sphere.

10 'PARA' is a parallelepiped. It has 6 parameters, the half length in x, the half length in y, the half length in z, the angle w.r.t. the y axis from the centre of the low y edge to the centre of the high y edge, and the theta phi polar angles from the centre of the low z face to the centre of the high z face.

11 'PGON' is a polygon. It has at least 10 parameters, the lower phi limit, the range in phi, the number of straight sides (of equal length) between those phi limits, the number of z planes (at least two) where the distances to the z axis are changing, z coordinate of first plane, the shortest distances RMIN & RMAX from the z axis to the inside straight edge and the outside straight edge for the first plane, Z, RMIN, RMAX for the second plane, and so on.

12 'PCON' is a polycone. It has at least 9 parameters, the lower phi limit, the range in phi, the number (at least two) of z planes where the radius is changing for each z boundary and the z coordinate, the minimum radius and the maximum radius.

28 'GTRA' general twisted trapezoid.

Essentially it is a TRAP shape, except that it is twisted in the x,y plane as a function of z. At z= +/− DZ the parallel sides are inclined to the x axis by +/− an angle TWIST, which is one of the parameters. The shape is defined by the eight corners and is assumed to be constructed of straight lines joining points on the boundary of the trapezium at z= −DZ to the corresponding points on the boundary at z=DZ. So far, it is not allowed to divide volumes with shape GTRA.

GTRA Parameters

User Supplied:

DZ = Half length perpendicular to the parallel faces (Z axis).

TH = Theta (polar angle) of line joining centres of parallel faces.

PH = Phi (azimuth) of line joining centres of parallel faces.

TWIST = Twist angle of parallel sides at z= +/− DZ.

Y1 = Half distance between parallel sides at z= −DZ.

XL1 = Half length of low y parallel side at z= −DZ.

XH1 = Half length of high y parallel side at z= −DZ.

TH1 = Angle to perpendicular to parallel sides of line joining their centres at z= −DZ.

Y2 = Half distance between parallel sides at z=DZ.

XL2 = Half length of low y parallel side at z=DZ.

XH2 = Half length of high y parallel side at z=DZ.

TH2 = Angle to perpendicular to parallel sides of line joining their centres at z=DZ.

Internal:

First 12 are the user supplied parameters. Then there are an extra 18 which are as follows:

DX0DZ = DX/DZ for the line joining the centres of the faces at Z= +/− DZ.

DY0DZ = DY/DZ for the line joining the centres of the faces at Z= +/− DZ.

X01 = X at Z=0 for line joining the + on parallel side,
perpendicular corners at Z= +/− DZ.

Y01 = Y at Z=0 for line joining the + on parallel side,

+ on perpendicular corners at Z= + / − DZ.

DXDZ1 = DX/DZ for line joining the + on parallel side, + on perpendicular corners at Z= + / − DZ.

DYDZ1 = DY/DZ for line joining the + on parallel side, + on perpendicular corners at Z= + / − DZ.

X02 = X at Z=0 for line joining the − on parallel side, + on perpendicular corners at Z= + / − DZ.

Y02 = Y at Z=0 for line joining the − on parallel side, + on perpendicular corners at Z= + / − DZ.

DXDZ2 = DX/DZ for line joining the − on parallel side, + on perpendicular corners at Z= + / − DZ.

DYDZ2 = DY/DZ for line joining the − on parallel side, + on perpendicular corners at Z= + / − DZ.

X03 = X at Z=0 for line joining the − on parallel side, − on perpendicular corners at Z= + / − DZ.

Y03 = Y at Z=0 for line joining the − on parallel side, − on perpendicular corners at Z= + / − DZ.

DXDZ3 = DX/DZ for line joining the − on parallel side, − on perpendicular corners at Z= + / − DZ.

DYDZ3 = DY/DZ for line joining the − on parallel side, − on perpendicular corners at Z= + / − DZ.

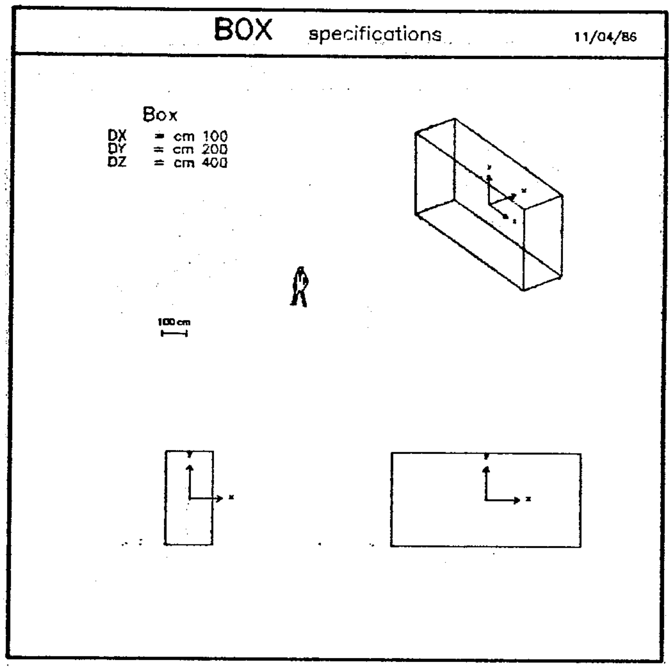
X04 = X at Z=0 for line joining the + on parallel side, − on perpendicular corners at Z= + / − DZ.

Y04 = Y at Z=0 for line joining the + on parallel side, − on perpendicular corners at Z= + / − DZ.

DXDZ4 = DX/DZ for line joining the + on parallel side, − on perpendicular corners at Z= + / − DZ.

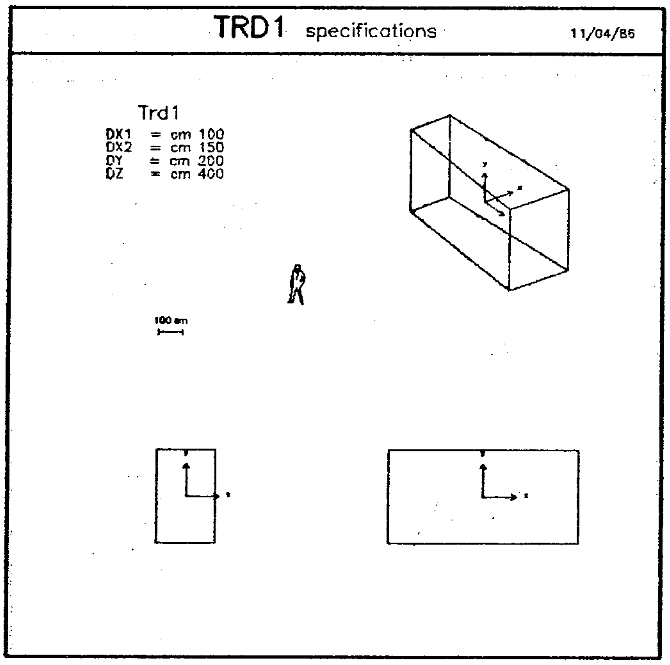
DYDZ4 = DY/DZ for line joining the + on parallel side, − on perpendicular corners at Z= + / − DZ.

Shapes BOX,TRD1,TRD2,TRAP



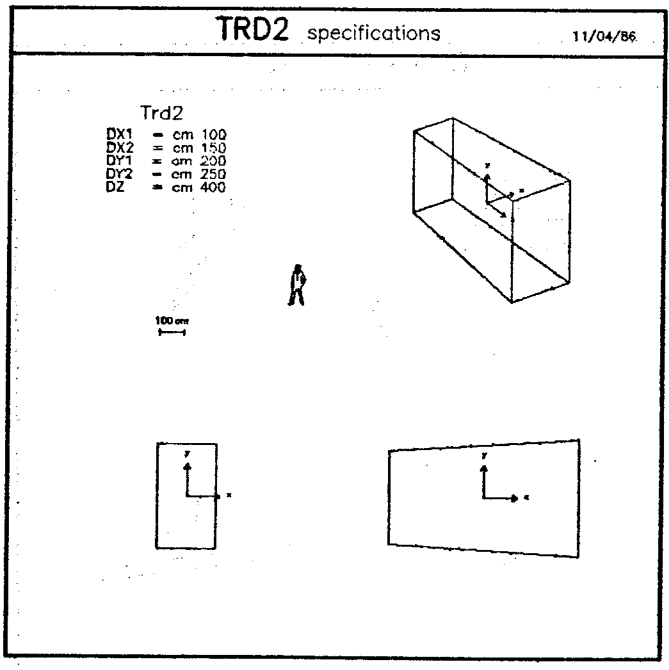
BOX specifications 11/04/86

Box
DX = cm 100
DY = cm 200
DZ = cm 400



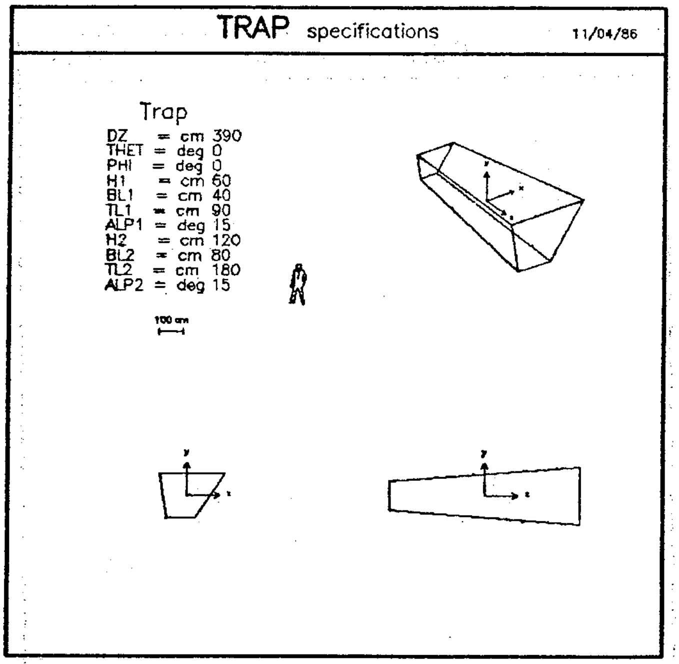
TRD1 specifications 11/04/86

Trd1
DX1 = cm 100
DX2 = cm 150
DY = cm 200
DZ = cm 400



TRD2 specifications 11/04/86

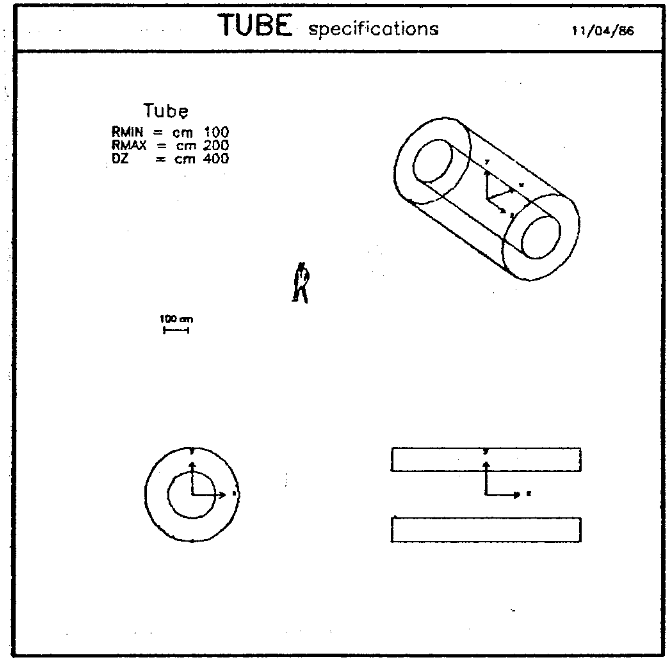
Trd2
DX1 = cm 100
DX2 = cm 150
DY1 = cm 200
DY2 = cm 250
DZ = cm 400



TRAP specifications 11/04/86

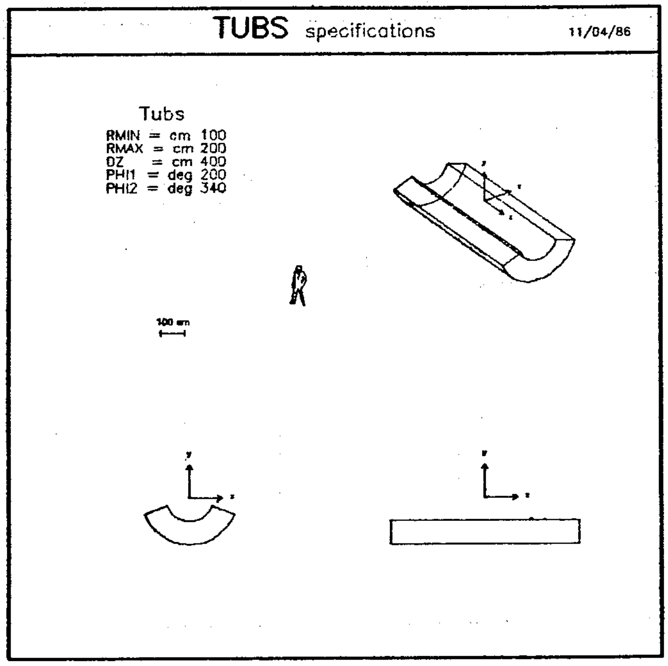
Trap
DZ = cm 390
THET = deg 0
PHI = deg 0
H1 = cm 60
BL1 = cm 40
TL1 = cm 90
ALP1 = deg 15
H2 = cm 120
BL2 = cm 80
TL2 = cm 180
ALP2 = deg 15

Shapes TUBE,TUBS,CONE,CONS



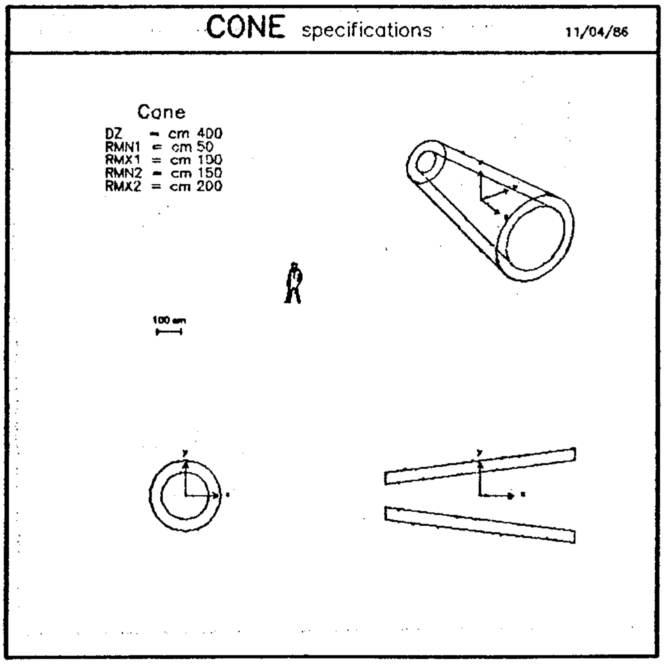
TUBE specifications 11/04/86

Tube
RMIN = cm 100
RMAX = cm 200
DZ = cm 400



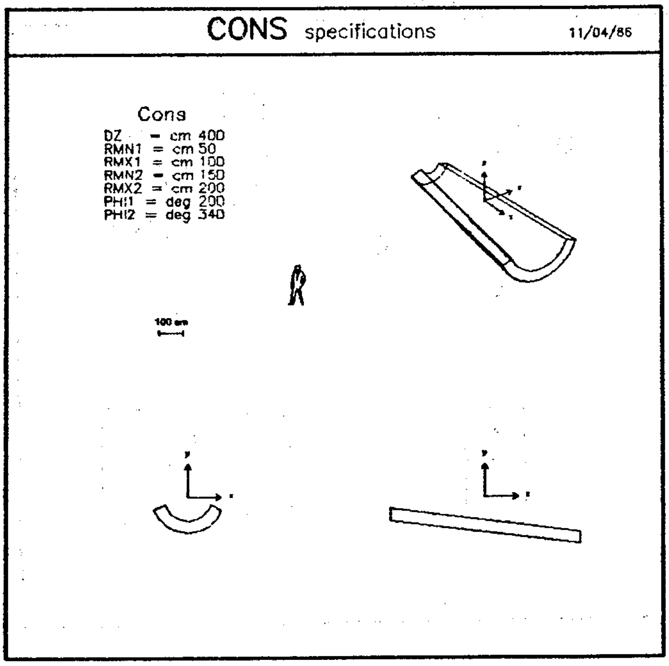
TUBS specifications 11/04/86

Tubs
RMIN = cm 100
RMAX = cm 200
DZ = cm 400
PHI1 = deg 200
PHI2 = deg 340



CONE specifications 11/04/86

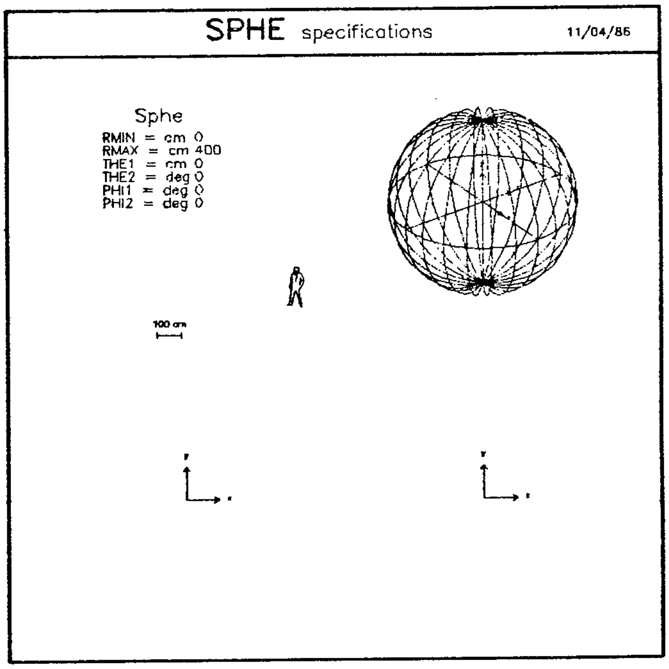
Cone
DZ = cm 400
RMN1 = cm 50
RMX1 = cm 100
RMN2 = cm 150
RMX2 = cm 200



CONS specifications 11/04/86

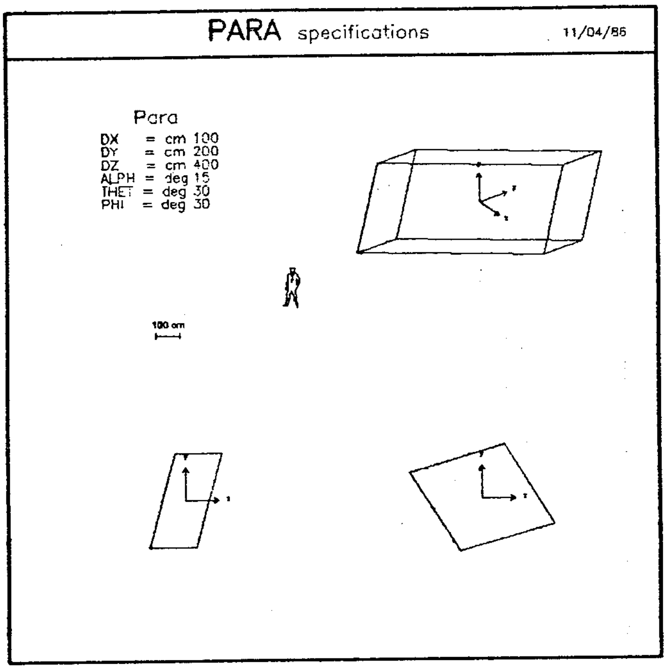
Cons
DZ = cm 400
RMN1 = cm 50
RMX1 = cm 100
RMN2 = cm 150
RMX2 = cm 200
PHI1 = deg 200
PHI2 = deg 340

Shapes PARA,SPHE,PGON,PCON



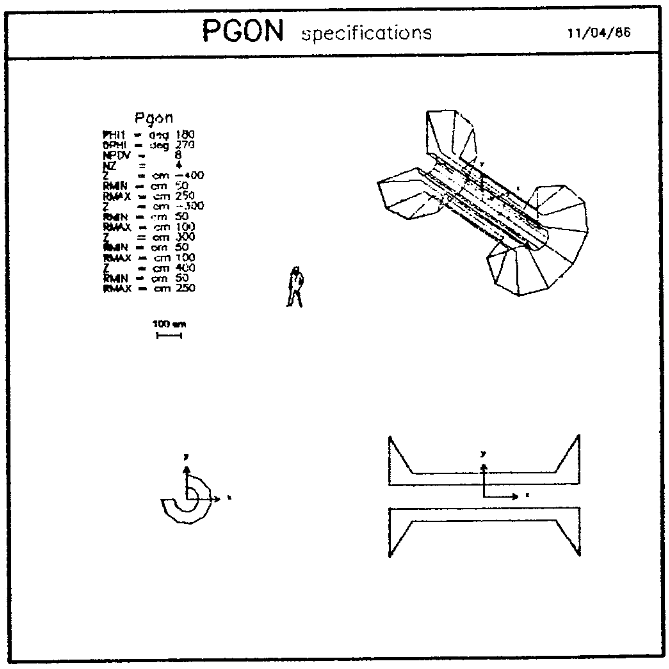
SPHE specifications 11/04/86

```
Sphe
RMIN  = cm  0
RMAX  = cm  400
THE1  = cm  0
THE2  = deg 0
PHI1  = deg 0
PHI2  = deg 0
```



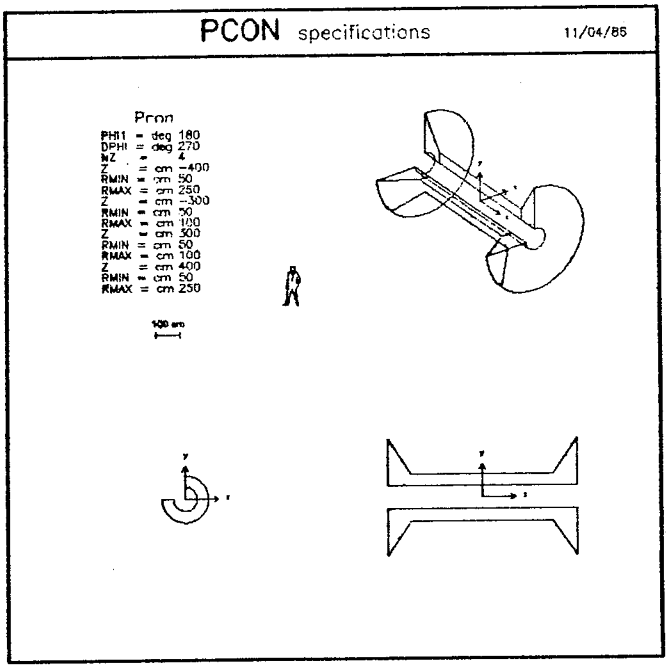
PARA specifications 11/04/86

```
Para
DX    = cm  100
DY    = cm  200
DZ    = cm  400
ALPH  = deg 15
THET  = deg 30
PHI   = deg 30
```



PGON specifications 11/04/86

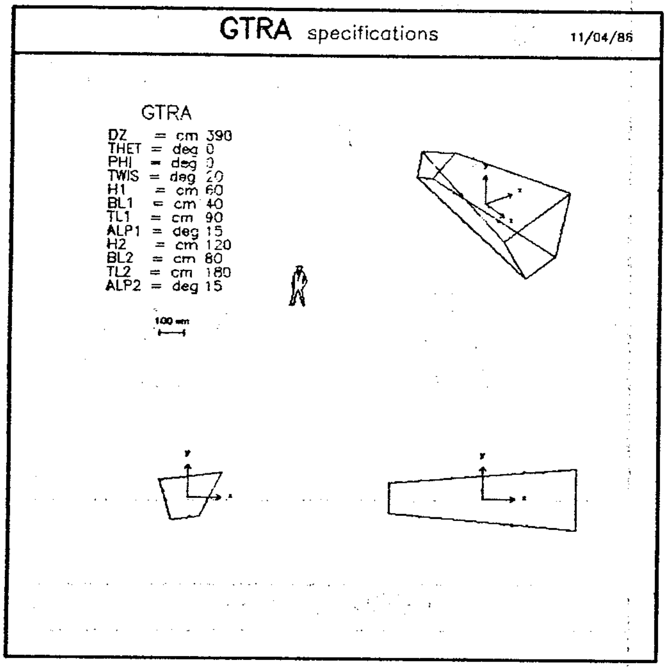
```
Pgon
PHI1  = deg 180
DPHI  = deg 270
NPDV  =     8
NZ    =     4
Z     = cm  -400
RMIN  = cm  50
RMAX  = cm  250
Z     = cm  -300
RMIN  = cm  50
RMAX  = cm  100
Z     = cm  300
RMIN  = cm  50
RMAX  = cm  100
Z     = cm  400
RMIN  = cm  50
RMAX  = cm  250
```



PCON specifications 11/04/86

```
Pcon
PHI1  = deg 180
DPHI  = deg 270
NZ    =     4
Z     = cm  -400
RMIN  = cm  50
RMAX  = cm  250
Z     = cm  -300
RMIN  = cm  50
RMAX  = cm  100
Z     = cm  300
RMIN  = cm  50
RMAX  = cm  100
Z     = cm  400
RMIN  = cm  50
RMAX  = cm  250
```

Shape GTRA



GTRA specifications 11/04/86

GTRA

```
DZ    = cm  390
THET  = deg 0
PHI   = deg 0
TWIS  = deg 20
H1    = cm  60
BL1   = cm  40
TL1   = cm  90
ALP1  = deg 15
H2    = cm  120
BL2   = cm  80
TL2   = cm  180
ALP2  = deg 15
```

Author(s) : R.Brun
Origin : R.Brun, A.McPherson

Submitted: 15.08.83
Revised: 28.04.86

Creation of a Volume

CALL GSVOLU(NAME,SHAPE,NMED,PAR,NPAR,IVOLU*)

Defines a volume with a given name, shape, tracking medium number and shape parameters.

There is a way for the user to indicate that dimensions are to be chosen automatically by the supplied routines. If the user specifies a dimensional parameter to be negative, then, in each case where the user places this volume within another, this dimension will be chosen to be the maximum that will fit inside the mother volume. This facility is of principal use in conjunction with the slice division of volumes using GSDVN, GSDVT, GSDVX. It is implemented only in the absence of a relative rotation and only for rectangles and trapezoids within each other and for cylinders and cones or segments thereof within each other in the case of no offset in x or y.

NAME Unique 4-character name.

SHAPE 4-character name of a system supplied shape.

NMED Tracking medium number for the region of the volume. If volumes are located within this volume then this value will be overridden by the lower level definition.

PAR Array containing the shape parameters.

NPAR Number of such parameters. If this is zero then the parameters will be expected at positioning time through calls to GSPOSP [GEOM 120].

IVOLU Number returned by the subroutine as system volume number
if IVOLU≤0, then there is an error in the call.

Examples

```
DIMENSION PECAL(6)
DATA PECAL/1.71,4.,0.2,3*0./

CALL GSVOLU('ECAL','BOX ',1,PECAL,6,IVOLU)
```

CALL GPVOLU(IVOLU)

Prints the volume parameters

IVOLU For volume IVOLU.
(for all volumes if IVOLU = 0)

Author(s) : R.Brun, A.McPherson
Origin : Same

Submitted: 15.08.83
Revised: 14.05.86

Positioning a Volume inside its Mother – Case 1

```
CALL GSPOS(NAME,NR,MOTHER,X,Y,Z,IROT,KONLY)
```

Places a copy of the volume NAME (previously defined by a call to GSVOLU) with user number NR inside its mother volume MOTHER.

GSPOS can be called several times with the same name, to place copies of the same volume (and of its contents) in various positions, either in the same mother volume or in several different ones. The data structure is so designed that, in any case, the substructure of a volume is defined and stored only once. Reference is always made to the first one wherever the volume is encountered.

NAME A 4-character volume name (must have been defined by a call to GSVOLU).

NR Copy number of the volume NAME being placed.

MOTHER The 4-character mother name of this volume (i.e. the volume within which this volume is to be placed). The master reference system is defined by the top volume which does not need to be positioned at all.

X,Y,Z Position of the volume in the mother reference system.

IROT Rotation matrix number describing the orientation of the volume relative to the coordinate system of the mother (see GSROTM).

KONLY A 4-character flag to indicate, at search time, whether a point found to be in this volume may also be in other volumes which are not direct descendants of it. If it is set to 'ONLY' then the search routines will assume that, when in it, the only further searching required is in its contents and their contents etc. The search will not look at other volumes at the same or higher levels or in separate branches. If it is set to 'MANY' the search will include other volumes which are contents of the same parent and if the parent is also 'MANY' the search will include volumes at that level and so on until an 'ONLY' volume is reached. Eventually, a list of 'MANY' type volumes which are descendants of the last (lowest) 'ONLY' type volume will be generated. Of these the lowest level of search will be taken and if there are several at that level the first of them which was found will be taken.

Author(s) : R.Brun, F.Bruyant, A. McPherson
Origin : R.Brun, F.Bruyant, M.Maire

Submitted: 15.08.83
Revised: 28.04.86

Positioning a Volume inside its Mother – Case 2

```
CALL GSPOSP(NAME,NR,MOTHER,X,Y,Z,IROT,KONLY,PAR,NPAR)
```

Places a copy of the volume NAME (previously defined by a call to GSVOLU with NPAR = 0) with user number NR inside its mother MOTHER.

GSPOSP can be called several times with the same name, to place copies of the same volume (and of all its contents) with various parameters in various positions, either in the same mother volume or in several different ones. The data structure is so designed that, in any case, the substructure of a volume is defined and stored only once. Reference is always made to the first one wherever the volume is encountered.

NAME A 4-character volume name (must have been defined by a call to GSVOLU).

NR Copy number of the volume NAME being placed

MOTHER The 4-character mother name of this volume (i.e. the volume within which this volume is to be placed). The master reference system is defined by the top volume which does not need to be positioned at all.

X,Y,Z Position of the volume in the mother reference system.

IROT Rotation matrix number describing the orientation of the volume relative to the coordinate system of the mother (see routine GSROTM).

KONLY A 4-character flag to indicate, at search time, whether a point found to be in this volume may also be in other volumes which are not direct descendants of it. If it is set to 'ONLY' then the search routines will assume that, when in it, the only further searching required is in its contents and their contents etc. The search will not look at other volumes at the same or higher levels or in separate branches. If it is set to 'MANY' the search will include other volumes which are contents of the same parent and if the parent is also 'MANY' the search will include volumes at that level and so on until an 'ONLY' volume is reached. Eventually, a list of 'MANY' type volumes which are descendants of the last (lowest) 'ONLY' type volume will be generated. Of these the lowest level of search will be taken and if there are several at that level the first of them which was found will be taken.

PAR Array containing the parameters.

NPAR Number of parameters.

Author(s) : R.Brun, A.McPherson
Origin : Same

Submitted: 29.09.83
Revised: 28.04.86

Division of a Volume into a given Number of Cells

```
CALL GSDVN(NAME,MOTHER,NDIV,IAXIS)
```

A new volume will be created with the name NAME and the appropriate parameters to define the slice shape.

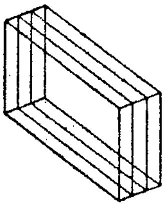
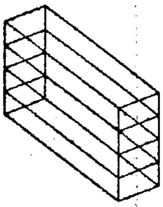
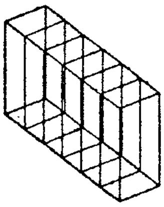
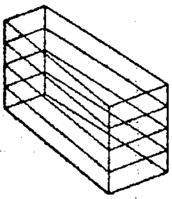
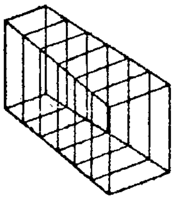
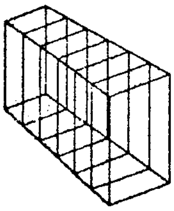
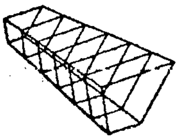
Local coordinate systems will be defined parallel to the system in which the slicing is defined and displaced to the geometric centre of each slice.

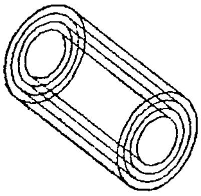
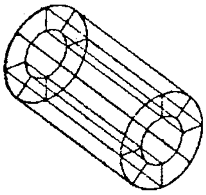
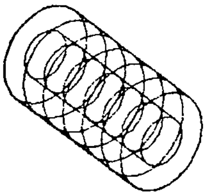
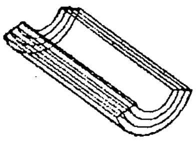
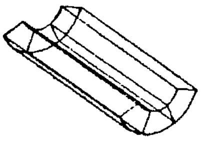
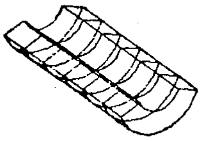
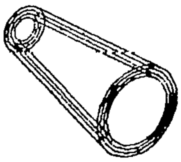
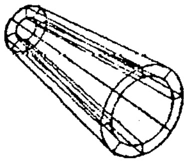
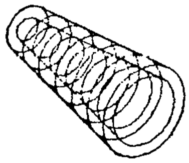
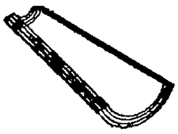
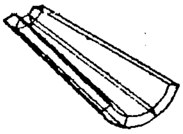
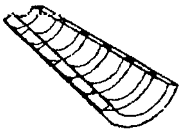
If the mother volume has cylindrical or conical shape, IAXIS = 1 will generate slices in the radius (producing concentric tubes or cones), and IAXIS = 2 will generate slices in phi (producing segments). In this latter case the local coordinate systems will be rotated so that phi = 0 passes through the centre of the segment. If a cylindrical shape is sliced with IAXIS = 3, the usual linear slicing will be done and the coordinate systems will be generated by displacements along the Z axis to the Z centre of each slice.

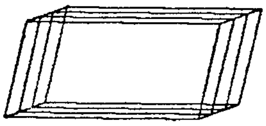
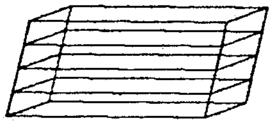
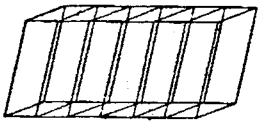
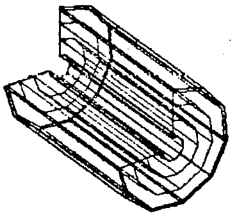
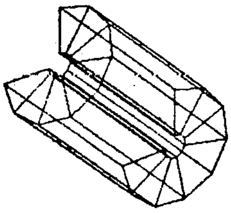
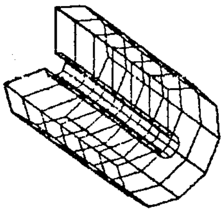
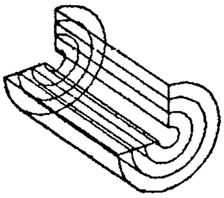
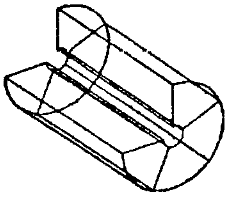
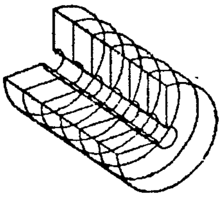
This routine allows the user to generate a large number of identical volumes filling a defined volume. Further, if the slices cut a varying cross-section of the mother, then, by using negative values for the appropriate parameters of any volume to be positioned in the slice, the system will generate similar volumes - the same size in one dimension but of differing sizes in one or both of the other dimensions. This would be used, for example, to fill a trapezoidal tower with plates of constant thickness but of varying cross-section to match the varying cross-section of the tower.

Clearly a single mother volume can be subdivided, either by the division technique (e.g. GSDVN) or by positioning other volumes within it (e.g. GSPOS), but not by both techniques simultaneously.

NAME	A unique 4-character name for the volume to be generated by subdivision of the volume called MOTHER.
MOTHER	The 4-character name of the volume that has to be subdivided
NDIV	Number of divisions into which the mother volume is to be sliced.
IAXIS	Gives the axis to be sliced up.

IAXIS SHAPE	1	2	3
BOX			
TRD1			
TRD2			
TRAP			

IAXIS SHAPE	1	2	3
TUBE			
TUBS			
CONE			
CONS			

IAXIS SHAPE	1	2	3
SPHE			
PARA			
PGON			
PCON			

Author(s) : A.McPherson
Origin : F.Bruyant, A.McPherson, M.Maire

Submitted: 17.12.83
Revised: 20.05.86

Division of a Volume into Cells of a given Size

```
CALL GSDVT(NAME,MOTHER,STEP,IAXIS,NUMED,NDVMX)
```

A new volume will be created with the name NAME and the appropriate parameters to define the slice shape. The full range of the mother will be divided in sections of the user supplied step. If the step does not fit to form an exact number of divisions, then the largest number which will fit will be made and any excess space will be equally divided between each end of the range of the mother and these extra spaces will be assumed to be just the mother.

NAME	A unique 4-character name for the volume to be generated by subdivision of the volume called MOTHER.
MOTHER	The 4 character name of the volume that has to be subdivided.
STEP	Thickness of divisions into which the mother volume is to be sliced.
IAXIS	Gives the axis to be sliced up.
NUMED	Medium number for the divisions. This can be different from that of the mother in order to account for the spaces left at each end of the divided range when the step does not divide the mother exactly. If NUMED ≤ 0 the medium of the mother is used by default
NDVMX	Expected (maximum) number of divisions, or 0 (interpreted as 255)

See also GSDVN write-up.

Calls to GSDVS are identical to calls to GSDVT with NDVMX = 0 . In the next program version GSDVS will not be supported anymore.

Author(s) : F.Bruyant, A.McPherson
Origin : R.Brun, F.Bruyant, A.McPherson

Submitted: 17.12.83
Revised: 20.05.86

Division of a Volume – General case

```
CALL GSDVX(NAME,MOTHER,NDIV,IAXIS,STEP,C0,NUMED,NDVMAX)
```

Divides the volume MOTHER into NDIV divisions of size STEP along axis IAXIS starting at coordinate C0. See also GEOM 130 and 140.

NDIV (or STEP) can be set negative or 0, in which case it will be computed from the MOTHER's size. The case with both NDIV and STEP positive is not coded yet. It will permit to leave gaps at both ends of the MOTHER. Provisionally the code consists of a call to either GSDVN2 or GSDVT2.

```
CALL GSDVN2(NAME,MOTHER,NDIV,IAXIS,C0,NUMED)
```

A new volume will be created with the name NAME and the appropriate parameters to define the slice shape. The slices will commence at the user specified coordinate value and extend to the end of the volume. This range from origin to upper coordinate limit of the mother volume will be divided exactly into the user supplied number of slices. In the case of a phi division of a complete tube or cone the whole 360 degrees will be divided into the user supplied number of slices no matter what the origin is. The origin, in this case, just moves the division boundaries. In all other cases the search routines will assume a point is just in the mother if the coordinate value is less than the value of the user supplied origin.

NAME A unique 4-character name for the volume to be generated by subdivision of the volume called MOTHER.

MOTHER The 4-character name of the volume that has to be subdivided

NDIV Number of divisions into which the mother volume is to be sliced.

IAXIS Gives the axis to be sliced up.

C0 Origin for the divisions.

NUMED Medium number for the divisions. This can be different from that of the mother to account for the case where the origin does not correspond to the lower edge of the parent. In this case the medium number of the parent will be returned for that region of it with coordinate values less than the origin, except in the case of phi divisions of a complete tube or cone which is discussed above.

```
CALL GSDVT2(NAME,MOTHER,STEP,IAXIS,C0,NUMED,NDVMX)
```

A new volume will be created with the name NAME and the appropriate parameters to define the slice shape. The slices will commence at the user specified coordinate value and extend to the end of the volume. This range from origin to upper coordinate limit of the mother volume will be divided in sections of the user supplied step. If the step does not fit to form an exact number of divisions, then the largest number which will fit will be made and any excess space up to the end of the mother volume will be assumed to be just the mother. In the case of a phi division of a complete tube or cone the region to be divided will be assumed to the complete circle from origin back to itself. In all other cases the search routines will assume a point is just in the mother if the coordinate value is less than the value of the user supplied origin.

NAME	A unique 4-character name for the volume to be generated by subdivision of the volume called MOTHER.
MOTHER	The 4-character name of the volume that has to be subdivided
STEP	Thickness of divisions into which the mother volume is to be sliced.
IAXIS	Gives the axis to be sliced up.
C0	Origin for the divisions.
NUMED	Medium number for the divisions. This can be different from that of the mother to account for the case where the origin does not correspond to the lower edge of the parent or where the step does not divide the remaining range exactly. In either case the medium number of the parent will be returned for regions outside the divisions.
NDVMX	Expected (maximum) number of divisions, or 0 (interpreted as 255)

Calls to GSDVS2 are identical to calls to GSDVT2 with NDVMX = 0 . In the next program version GSDVS2 will not be supported anymore.

Author(s) : R.Brun, F.Bruyant, A.McPherson
Origin : Same

Submitted: 01.11.83
Revised: 28.04.86

The Volume data structure JVOLUM

ISEARC search flag [GEOM 400, GEOM 410]
0 initial GSPOS order (default), or option GSNEXT
−1 binary search (GSORD)
1 statistical ordering (GSSEAR)
3 user ordering

ISHAPE system shape number

NIN number of volumes imbedded in the mother volume. If it is negative then the volume is divided into slices

NMED medium number for the volume

NPAR number of shape parameters

NATT number of attributes

PAR array of shape parameters

IAT array of attributes

in the case of slice division (NIN < 0):

IAXIS defines the direction of the slices (1,2,3)

IVO system volume number

NDIV number of slices (may be 0, or −NDVMX, if computed dynamically)
(NDVMX, argument of GSDVT)

C0 minimum coordinate limit

STEP coordinate step from slice to slice

in the case of object insertion (NIN > 0):

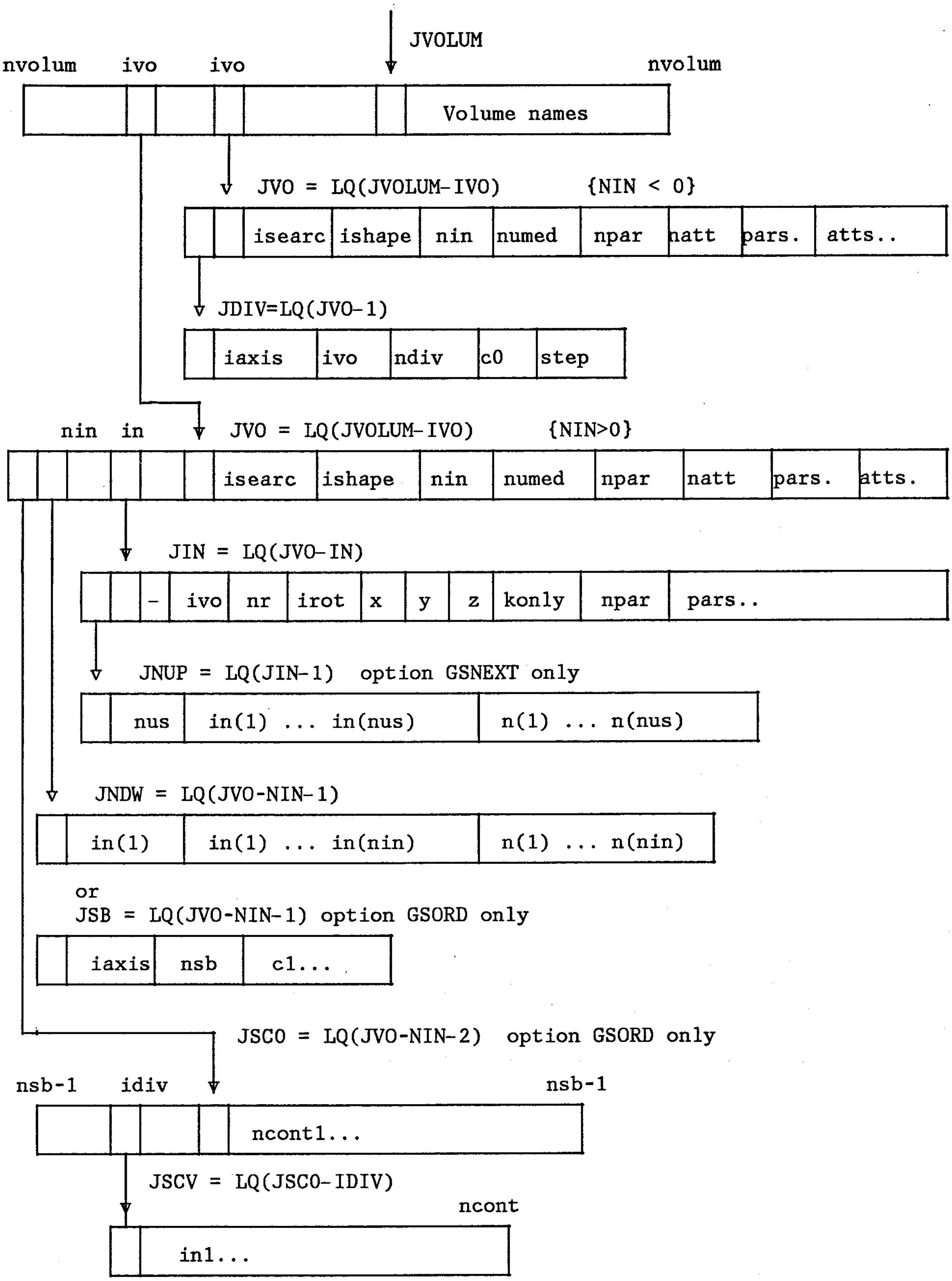
IVO system volume number

NR user number

IROT rotation matrix number defining the orientation of the volume

X,Y,Z defines the position of the volume

KONLY indicates whether it is sufficient to find a point within this volume or whether there may be some ambiguity with other volumes at the same level


JVOLUM
nvolum
ivo
ivo
nvolum
Volume names
JVO = LQ(JVOLUM-IVO)
{NIN < 0}
isearc
ishape
nin
numed
npar
natt
pars.
atts..
JDIV=LQ(JVO-1)
iaxis
ivo
ndiv
c0
step
nin
in
JVO = LQ(JVOLUM-IVO)
{NIN>0}
isearc
ishape
nin
numed
npar
natt
pars.
atts.
JIN = LQ(JVO-IN)
-
ivo
nr
irot
x
y
z
konly
npar
pars..
JNUP = LQ(JIN-1)   option GSNEXT only
nus
in(1) ... in(nus)
n(1) ... n(nus)
JNDW = LQ(JVO-NIN-1)
in(1)
in(1) ... in(nin)
n(1) ... n(nin)
or
JSB = LQ(JVO-NIN-1) option GSORD only
iaxis
nsb
c1...
JSCO = LQ(JVO-NIN-2)   option GSORD only
nsb-1
idiv
nsb-1
ncont1...
JSCV = LQ(JSCO-IDIV)
ncont
in1...

Author(s) : R. Brun, F.Carena
Origin : GEANT2

Submitted: 01.6.83
Revised: 30.05.86

Handling Rotation Matrices

CALL GSROTM(IROT,THETA1,PHI1,THETA2,PHI2,THETA3,PHI3)

Computes and stores the rotation matrix with serial number IROT in the data structure JROTM. (See notations and conventions in BASE 090).

THETA1	polar angle for axis I
PHI1	azimuthal angle for axis I
THETA2	polar angle for axis II
PHI2	azimuthal angle for axis II
THETA3	polar angle for axis III
PHI3	azimuthal angle for axis III

Note : The angles THETA and PHI must be given in degrees.

Example 1 :

Axis I parallel to 1		THETA1 = 90.	PHI1 = 0.
Axis II parallel to 2	----->	THETA2 = 90.	PHI2 = 90.
Axis III parallel to 3		THETA3 = 0. PHI3 = 0.	

Example 2 :

Axis I parallel to 2		THETA1 = 90.	PHI1 = 90.
Axis II parallel to 3	----->	THETA2 = 0. PHI2 = 0.	
Axis III parallel to 1		THETA3 = 90.	PHI3 = 0.

CALL GPROTM(IROT)

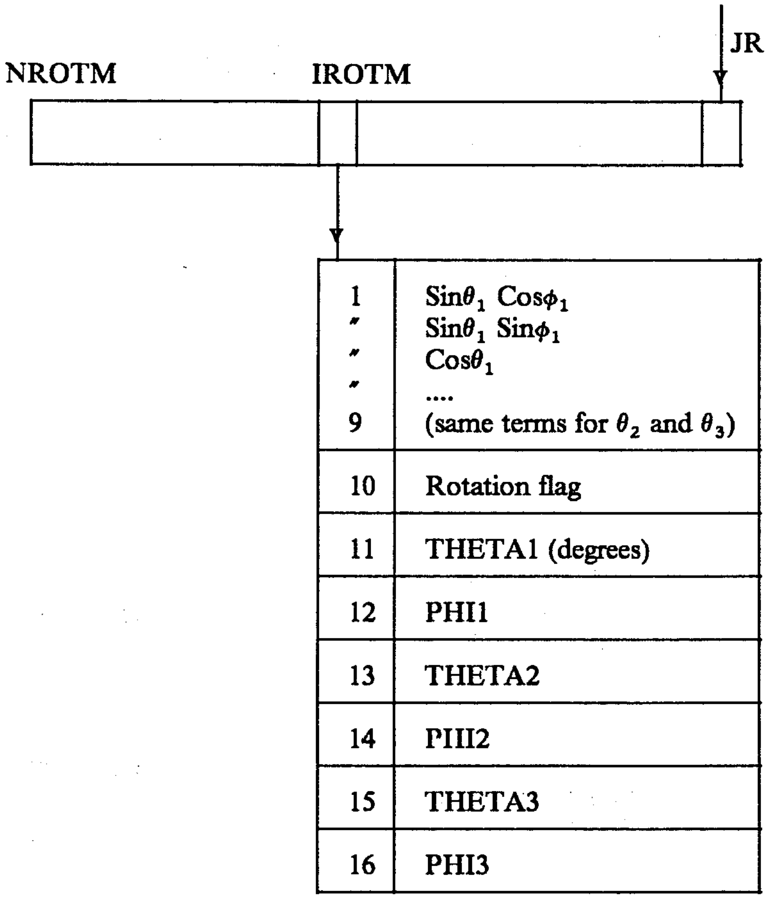
Prints the rotation matrix elements and angles

IROT for the rotation matrix IROT.
(for all matrices if IROT=0)

Author(s) :
Origin : GEANT2/3

Submitted: 01.11.78
Revised: 28.05.86

The Rotation Matrix data structure JROTM



1 " " " 9	$\mathrm{Sin}\theta_1$ $\mathrm{Cos}\phi_1$ $\mathrm{Sin}\theta_1$ $\mathrm{Sin}\phi_1$ $\mathrm{Cos}\theta_1$ .... (same terms for θ_2 and θ_3)
10	Rotation flag
11	THETA1 (degrees)
12	PHI1
13	THETA2
14	PIII2
15	THETA3
16	PHI3

JR = LQ(JROTM − IROTM) pointer to rotation matrix number IROTM

The rotation flag is computed internally to provide the possibility at execution time to quickly recognize simple rotation configurations.

Author(s) : R.Brun, A.C.McPherson, F.Bruyant
Origin : Same

Submitted: 18.12.83
Revised: 20.05.86

Finding Volumes which correspond to given Space Point

```
CALL GMEDIA(X,*NUMED*)
```

Searches the tree structure to find in which tracking medium the point X is. The tracking medium is returned in NUMED.

X array of dimension 3, giving a point in master coordinates.
NUMED should be set to 0 on input when starting a new track,
returns with the medium number. If this is zero, then the point is outside the detector.

GMEDIA uses the data structure established by the routines described above to conduct its search. It also uses the history of the last search in order to look first in the same object. Then it will look in the parent of that object and so on until it finds an object in which it is. Then it will look at the contents of this object and so on until it is in a unique object with no contents. If it does not find such but the downward search terminates in a 'MANY' object, then it will back track to the parent and look at other contents thereof. If it is in one it will resume the downward search. Eventually the lowest 'ONLY' object in which the point lies will be found. The contents of this will be exhaustively searched and the lowest level object within this 'ONLY' object and in which the point lies will be taken. If there are several at the same level then the first of these which was found will be taken. Various systems are available for controlling the search sequence through the routine GSEAR. Binary searches along an axis can be set up by GSORD. Nearest neighbour search patterns can be established by GSNEXT and statistically driven search ordering can be done with GSSEAR and GOSEAR.

The multi-level description is stored in the common block GCVOLU [GEOM 010].

GMEDIA updates the history before returning. If the user has some reason to restart at the top level, or any level higher than GMEDIA has set, only NLEVEL need be modified. For example, the top level can be specified by setting NLEVEL to 0 or 1. If the user wishes GMEDIA to start at some other place, then the whole history down to the appropriate level must be replaced by the user. This can be done directly or by use of the routine GLVOLU which, given the level to which it is to be filled and the names and numbers at each level up to that, will fill the common /GCVOLU/ appropriately.

```
CALL GMENEW(X,NUMED*,ISAME*)
```

Checks if point X is still in the same volume or not.

If yes, then ISAME = 1

If not, ISAME = 0 In that case GMENEW is equivalent to GMEDIA and NUMED returns the tracking medium number of the new volume.

CALL GINVOL(VECT,VOUT,ISAME*)

Checks that point VOUT is in the same volume as VECT. If so then returns ISAME = 1, otherwise returns ISAME = 0.

Before calling this routine the common /GCVOLU/ must be correctly filled for example through a call to GMEDIA.

Author(s) : R.Brun, A.C.McPherson
Origin : Same

Submitted: 01.06.83
Revised: 20.05.86

Finding Distance to Next Boundary

CALL GNEXT(X,SNEXT*,SAFETY*)

Finds distance to the next boundary. It takes explicit account of shape content and uniqueness.

X Vector of current coordinates and direction cosines
SNEXT Distance from current point to volume boundary in specified direction
SAFETY Safety distance, irrespective of direction

The distance to the next boundary is evaluated analytically using the volume geometrical parameters.

When used in a tracking context it happens very frequently that a particle may have many steps in the same volume. In that case it is time consuming to always call the GMEDIA and GNEXT routines. The safety radius parameter can be exploited by the tracking routines, in such a way that GMEDIA and GNEXT are only called when the sum of all steps in the current volume is greater than SAFETY.

In some volumes with complicated configuration the evaluation of the safety radius will be much too time consuming. In order to not lose time on one side to gain it somewhere else, the safety radius is returned = 0.

CALL GNEXTS(X,SH,P,*IDIST*,SNEXT*,SAFETY*)

This is a steering routine which calls one of a series of specialized routines depending on the shape of the volume for which it is called. These routines compute both the absolute shortest distance from the point (assumed to be INSIDE the volume) to the boundary of the volume and also the distance along a vector to that boundary. If a flag is zero or if the distance along the vector is less than the current value of the argument SNEXT then that argument is updated with the new value. The absolute shortest distance is always returned.

X An array, the first 3 components of which define the point and the second three define the vector.
SH Indicates the shape number of the volume [GEOM 050].
P An array containing the parameters defining the volume.
IDIST Is the flag indicating whether the value of SNEXT should be overwritten anyway (IDIST = 0) or only with a lower value (IDIST = 1). It is always returned as 1.
SNEXT When overwritten (see IDIST) the distance along the vector to the boundary of the volume.
SAFETY Always the shortest distance to the boundary.

CALL GNSOUT(X,SH,P,*IDIST*,SNEXT*,SAFETY*)

This is a steering routine which calls one of a series of specialized routines depending on the shape of the volume for which it is called. These routines compute both the absolute shortest distance from the point (assumed to be OUTSIDE the volume) to the boundary of the volume and also the distance along a vector to that boundary. If a positive value is found for the distance along the vector and if a flag is zero or if that distance is less than the current value of the argument SNEXT then that argument is updated with the new value. The absolute shortest distance is always returned.

Parameters explained in GNEXTS above, except that in the event that IDIST is zero and no positive distance to the volume along the vector is found then IDIST remains as zero.

Author(s) : R. Brun
Origin : Same

Submitted: 18.08.87
Revised:

Reference System Transformations

CALL GMTOD(XM,XD,IFLAG)*

Assuming the common /GCVOLU/ properly filled for at least the first NLEVEL levels, and given either the coordinates XM (IFLAG=1) or the direction cosines XM (IFLAG=2) in the Master Reference System, returns them in XD in the Local Reference System of the volume at level NLEVEL.

/GCVOLU/ is initialized either by the Tracking routines and GMTOD can then be called from GUSTEP, or by a call to GMEDIA (XM,NUMED), or by a call to GLVOLU (NLEVEL,NAMES,NUMBER,IER)

CALL GDTOM(XD,XM,IFLAG)*

Inverse operation to GMTOD.

Author(s) : A.C. McPherson
Origin : R.Brun, F.Bruyant

Submitted: 16.12.83
Revised: 18.03.86

Pseudo-Division of a Mother Volume

CALL GSORD(NAME,ICORD)

Computes the limits of each of the contents of a volume in a coordinate axis defined by the user. It then orders these coordinates and computes which contents are in each of the sections defined by neighbouring coordinates. The JVOLUM data structure for the volume is extended with this information in order to support a binary search in this coordinate for the contents of the volume. The search flag in this structure is set to -1 (= binary search). The coordinates are in the local system of the volume. At present, 3 cartesian coordinates, the radial coordinate of a cylindrical system with axis = z axis, the radial coordinate of a spherical system, the azimuthal (x,y plane) angular coordinate and the polar (axis = z axis) angular coordinate are supported.

NAME Name of parent volume.
ICORD Defines the coordinate type:
1 = x,
2 = y,
3 = z,
4 = R cylindrical (in the x,y plane),
5 = R spherical,
6 = Phi azimuth in the x,y plane, and
7 = Theta the polar angle with respect to the z axis.

Author(s) : F.Bruyant, A.C.McPherson
Origin : Same

Submitted: 16.12.83
Revised: 07.09.87

Ordering the Contents of a Volume

CALL GSNEXT(MOTHER,IN,NLIST,LIST)

Only for volumes with search flags = 0 or 1.

Stores a given ordered 'LIST' of NLIST daughter volumes to search for when leaving the IN'th daughter of a mother volume MOTHER.

−−−−−−−−> Default facility :

If IN = 0, for each content in turn, GSNEXT builds a list limited to the contents IN + 1 (if it exists), IN − 1 (if it exists) and IN itself.

CALL GSSEAR(MOTHER)

Requests statistical ordering for the contents of a given mother volume.
Set search flag to 1.

MOTHER Name of the mother volume.

CALL GOSEAR

For the contents of every volume for which an explicit request has been made through GSSEAR, this routine reorders the search lists according to the statistics so far collected and prints the corresponding information.

CALL GSUSEA(NAME,NIM)

Specifies user ordering for the contents of a given volume and informs the system as to the total number of contents it contains. Not yet operational.

Sets search flag to 3

Requires GUSEAR (called by GSEAR and GMEDIA) and GUNCON (called by GNEXT) to be coded by the user.

GEANT 3.10 USER'S GUIDE GEOM 500

Author(s) : R.Brun, P.Zanarini
Origin : Same

Submitted: 15.08.83
Revised: 22.04.86

Volume Attributes

```
CALL GSATT(NAME,IAT,IVAL)
```

Changes the attribute IAT of the volume called NAME to the value IVAL.

Attribute # 1	'WORK'	0 = Volume inactive 1 = active
Attribute # 2	'SEEN'	0 = unseen 1 = seen − 1, − 2 = (un)seen and tree unseen
Attribute # 3	'LSTY'	line style (1,2,3,4,..,etc.)
Attribute # 4	'LWID'	line width (1,2,3,4,5)
Attribute # 5	'COLO'	color code (1,2,3,4,5,6,7,8)
Attribute # 6	'FILL'	area filling with pattern (0,1,2,..,etc.)
Attribute # 7	'SET '	set number associated to this volume
Attribute # 8	'DET '	detector number associated to this volume
Attribute # 9	'DTYP'	detector type (1,2)
Attribute #10	'NODE'	< > 0 = a node is created for 3D (PIONS,etc.)

NAME	Volume name
IAT	Name of the attribute to be set
IVAL	Value to which the attribute is to be set.

```
CALL GFATT(NAME,IAT,IVAL*)
```

Returns in IVAL the attribute IAT of the volume called NAME. For the list of attributes see routine GSATT.

NAME	Volume name
IAT	Name of the attribute to be selected
IVAL	Attribute value returned.

```
CALL GFPARA(NAME,NR,INTEXT,NPAR*,NATT*,PAR*,ATT*)
```

Returns parameters PAR(1..NPAR) and attributes ATT(1..NATT) for the volume NAME with user number NR. INTEXT is used to select internal or external parameters.

NAME	Volume name
NR	User number

INTEXT	Internal (INTEXT = 1) or user parameter requested
NPAR	Number of parameters
NATT	Number of attributes
PAR	Array of parameters
ATT	Array of attributes.

Author(s) : F.Bruyant, A.McPherson
Origin : A.McPherson

Submitted: 16.12.83
Revised: 30.05.86

User Initialisation of the common block /GCVOLU/

CALL GLVOLU(NLEV,LNAM,LNUM,IER*)

Fills the geometrical parameters in common /GCVOLU/ for a volume at level NLEV in the physical tree, specified by the lists LNAM and LNUM of volumes NAMES and NUMBERs, and for all its ascendants up to the top level 1.

NLEV number of levels to consider
LNAM list of NLEV volumes NAMES
LNUM list of NLEV volume copy NUMBERs (or of cell numbers for a volume obtained by division)
IER Returns non-zero in case of user error
(A diagnostic is also printed in this case)

GLVOLU replaces the subroutine GSCMED which will be removed for the next release.

Note: This routine may be used in specific user analysis programs, outside the usual framework of the tracking. If so, the user has to initialize to zero the common /GCVOLU/.

Author(s) : R. Brun
Origin : Same

Submitted: 16.12.83
Revised: 28.04.86

Medium Search Statistics

CALL GBSTAT

Creates the data structure JGSTAT in order to accumulate statistics for the number of calls to routines GMEDIA,GNEXT,GINVOL and the total number of steps per volume. This routine is automatically called at initialisation time when the data card STAT is given.

CALL GPSTAT

Prints the volume statistics accumulated during the run. The table which is printed is useful for optimizing the tracking medium parameters associated to each volume. This routine is automatically called by GLAST when the data card STAT is given.

CALL GFSTAT

Fills banks for Volume statistics. This routine is automatically called by the GEOM routines when the data card STAT is given.

Author(s) : F.Bruyant, A.C.McPherson
Origin : A.C.McPherson

Submitted: 16.12.83
Revised: 20.05.86

End of Geometry Initialisation

CALL GGCLOS

This routine should be called after all volumes and positions have been defined. Its action is confined to creating and initializing the bank of initial search order and initial search statistics for those objects which have not had their contents ordered by GSORD.

It also calls the routine GGDETV which prepares the prototype lists of volume names and maximum multiplicities which permit to identify uniquely any sensitive detector whose generic name has been declared through the routine GSDETV (and not GSDET) [HITS 001].

Author(s) : F. Bruyant
Origin : GEANT3

Submitted: 15.08.84
Revised: 20.08.87

Introduction to the Detector Response package

1. The detector response package

In the context of GEANT3:

- a **hit** is the user defined 'information' recorded at tracking time to keep track of the interaction between one particle and a given detector, and regarded as necessary to compute the digitisations later.
- a **digitisation** is the user defined 'information' simulating the response of a given detector element, after tracking of a complete event.

The detector response package consists of tools to store in, and retrieve or print from, the data structures JSET, JHITS and JDIGI the information relevant to the hits and digitisations. A few subroutines which may help the user to solve some of the usual problems of digitisation in simple detectors have been added to the package, e.g. the intersection of a track with a plane or a cylinder and the digitisation of conventional drift and MWP chambers.

For complex set-ups with different types of detectors the user has normally to define several types of hits and digitisations. In addition to the hits generated by all particles of the current event, computing the digitisations requires usually some information about the intrinsic characteristics and performance of the detectors.

The information to be recorded for the hits and digitisations is highly experiment dependent, therefore only a framework can be proposed to store it. The solution adopted here should be satisfactory for most of the applications. Feedback from the users is needed and will be welcome.

Two remarks can be made:

- the stability of the information to be stored is usually reached much earlier for the hits than for the digitisations. Therefore the user may save computing time by designing an intermediate event output at the hits level.
- the scheme proposed for storing the digitisations should in any case be considered as an intermediate stage, a reshuffling of the data being necessary if the user wants to simulate more closely the specific format of the real data acquisition system.

2. SETs and DETectors

The reader is assumed to be familiar with the way the geometrical setup is described [GEOM 001], in particular with the concepts of logical volume structure and of physical path through the volume tree.

The user is required to classify into sets all sensitive detectors for which storing the hits in the data structure JHITS is wanted.

The 4-character names which identify the sets are user defined, and the list of sets which the user wants to activate for a given run can be entered through the data card SETS. The user is entirely free to group together, in one or in several sets, detectors of the same type or of different types. For convenience, it is recommended to have at least one set for each main component of the setup, e.g. hadron calorimeters, electromagnetic calorimeters, vertex chamber, etc.

A volume can be declared as sensitive detector through the tracking medium parameter ISVOL, and allocated to a set through the subroutine GSDET or GSDETV.

Each (logical) sensitive detector is identified by the 4-character name of the corresponding volume. As a given volume may describe several similar detectors of the physical setup, some additional information is needed for associating correctly the hits with the physical detectors.

When making use of GSDET the user has to enter the (shortest) list of volume ***names***, the vector NAMESV, which permits identification of the path through the physical tree, even in the presence of multiple copies at the volume level or at any lower level in the tree. The identification will be achieved when needed, by specifying a list of volume ***numbers***, the vector NUMBV, in one to one correspondence with the above list of volume names. This list, after packing, will constitute the identifier of the physical detector.

The routine GSDETV can be used i.o. GSDET and then the system (through GGDETV called by GGCLOS) constructs the lists NAMESV automatically and stores them in the structure JSET.

3. The basic user tools

The data structure JSET is built through calls to the routine GSDET or GSDETV(+GGDETV) which allocate detectors to sets and define their parameters, and to the auxiliary routines GSDETH, GSDETD and GSDETU which store respectively in the structure JSET, for each logical detector separately:

— the parameters required for the storage of the hit elements in the data structure JHITS, such as the packing and scaling conventions.
— the parameters required for the storage of the digitisations in the structure JDIGI, such as the packing conventions.
— the user parameters, which may consist, for instance, of the intrinsic detector characteristics needed for computing the digitisations.

To permit storage of more than one type of hit for a given sensitive detector, or to provide additional detector entries, detector 'aliases' can be defined through calls to the routine GSDETA. They are entered in the JSET structure as new detectors, with the same geometrical characteristics as the original one. Then, the user has the possibility to call appropriate routines GSDETH, GSDETD and GSDETU for this new detector.

During the tracking, for each step inside the sensitive detectors, under control of the subroutine ***GUSTEP***, the hits can be stored in the data structure JHITS with the subroutine GSAHIT (or GSCHIT, more appropriate for calorimetry). For each hit the information consists of:

— the reference to the track in the structure JKINE,
— the packed identifier of the physical detector, and
— the packed data for the different elements of the hit.

When the tracking has been completed for the whole event the digitisations can be computed in the user subroutine ***GUDIGI*** which may extract the hits with the subroutine GFHITS and store the digitisations in the data structure JDIGI, with the subroutine GSDIGI. For each digitisation the information should at least consist of:

- the reference to the track(s),
- the packed identifier of the physical detector, and
- the packed data for the digitisation itself.

4. Retrieval of geometrical information

The packed identifier of a physical detector, stored as part of the hit (or digitisation) information, is returned unpacked by the routines GFHITS (or GFDIGI) which extract the information from the JHITS or JDIGI structures, and may be used to retrieve the geometrical characteristics of the given detector.

This is automatized through the use of the routines GFPATH (which assumes that the sensitive detectors have been declared through the routine GSDETV, not GSDET) and GLVOLU which fills the common /GCVOLU/.

GFPATH prepares the lists LNAM and LNUM required by the routine GLVOLU [GEOM 001], according to the information preprocessed at initialisation time by the routine GGDETV and stored in the structure JSET.

GEANT 3.11 USER'S GUIDE **HITS 100**

Author(s) : R. Brun,F.Bruyant
Origin : Same

Submitted: 01.11.83
Revised: 20.08.87

Handling Sensitive DETector basic parameters

CALL GSDET(IUSET,IUDET,NV,NAMESV,NBITSV,IDTYPE,NWHI,NWDI,ISET,IDET*)*

Assigns a detector IUDET to the set IUSET and defines its basic parameters.

IUSET set identifier (4 characters), user defined
IUDET detector identifier (4 characters), has to be the name of an existing volume
NV number of volume descriptors
NAMESV vector of NV volume descriptors (4 characters)
NBITSV vector of NV bit numbers for packing the volume numbers
IDTYPA detector type, user defined
NWHI number of words for the primary allocation of HITS banks
NWDI number of words for the primary allocation of DIGI banks when first allocation not sufficient
ISET position of set in bank JSET
IDET position of detector in bank JS = LQ(JSET − ISET)

Remarks:

The vector NAMESV (length NV) contains the list of volume names which permit the identification of every physical detector with detector name IUDET. [See example in HITS 110].
Each element of the vector NBITSV (length NV) is the number of bits used for packing the number of the corresponding volume, when building the packed identifier of a given physical detector.
For more details see the example given in GSDETH.
The detector type IDTYPE is not used internally by GEANT and can be defined by the user to distinguish quickly between various kinds of detectors, in the routine GUSTEP for example.

CALL GSDETV(IUSET,IUDET,IDTYPE,NWHI,NWDI,ISET,IDET*)*

Arguments as in GSDET.
(NAMES, NBITSV will be computed by GGDETV called by GGCLOS)
See HITS 001.

CALL GFDET(IUSET,IUDET,NV,NAMESV*,NBITSV*,IDTYPE*)*

Returns the basic parameters for detector IUDET of set IUSET.

CALL GPSETS(IUSET,IUDET)

Prints SET and DETECTOR volume parameters.

IUSET User set identifier to be printed
if=0 prints all detectors of all sets

IUDET User detector identifier
if =0 prints all detectors of set IUSET.

Author(s) : F. Bruyant
Origin : Same

Submitted: 13.06.85
Revised: 20.08.87

Handling Detector Aliases

Detector 'aliases' can be specified for any sensitive detector for which the user either needs to store more than one type of hit or wants to define additional detector entries.

CALL GSDETA(IUSET,IUDET,IUALI,NWHI,NWDI,IALI)*

Defines an alias IUALI for detector IUDET of set IUSET. Enters it in the JSET structure as an additional detector in the corresponding set, at the position IALI. Copies to the link position IALI the GSDET parameter bank from the original detector IUDET, with empty links to the GSDETH, GSDETD and GSDETU parameter banks. The user can therefore call these three routines again with the arguments appropriate to the detector IUALI. Several aliases can be defined for the same detector through as many calls to GSDETA.

IUSET	user set identifier (4 characters)
IUDET	user detector identifier (4 characters), name of an existing volume
IUALI	user alias identifier (4 characters)
NWHI	number of words for the primary allocation of HITS banks
NWDI	number of words for the primary allocation of DIGI banks when first allocation not sufficient
IALI	position of alias in bank JS = LQ(JSET − ISET)

CALL GFDETA(IUSET,IUALI,IALI)*

Returns the position IALI of alias IUALI in set IUSET.

Author(s) : R. Brun
Origin : Same

Submitted: 01.11.83
Revised: 14.05.86

Handling Sensitive DETector Hit parameters

CALL GSDETH(IUSET,IUDET,NH,NAMESH,NBITSH,ORIG,FACT)

Defines hit parameters for detector IUDET of set IUSET.

IUSET	user set identifier
IUDET	user detector identifier
NH	number of elements per hit
NAMESH	the NH variable names for the hit elements
NBITSH	the NH bit numbers for packing the variable values
ORIG	The quantity packed in the structure JHITS for the Ith variable is a positive integer with NBITSH(I) bits and such that
FACT	IVAR(I) = (VAR(I)+ORIG(I))*FACT(I)

The routine is used at initialisation time once the geometrical volumes have been defined to describe the hit elements and the way to do packing in memory and on tape.

EXAMPLE

Assume an electromagnetic calorimeter ECAL divided into 40 PHI sections called EPHI. Each EPHI division is again divided along the Z axis in 60 objects called EZRI. Each EZRI is finally divided into 4 lead glass blocks called BLOC.

The geometrical information to describe one hit will then be:

The EPHI section number (between 1 and 40)
The EZRI division number (between 1 and 60)
The BLOC number (1 to 4)

The variables we want to store for each hit are for example:

X	x position of the hit in the lead glass block
Y	y
Z	z
E	energy of the particle at this point
ELOS	the energy deposited into this block

Example of one hit in that scheme:

```
EPHI 12
EZRI 41
BLOC  3
 X    7.89 cm
 Y    -345.6 cm
 Z    1234.8 cm
 E    12 Gev
 ELOS  11.85 Gev
```

The FORTRAN coding to define the set/det/hits information could be:

```
DIMENSION NAMESV(3),NBITSV(3)
DIMENSION NAMESH(5),NBITSH(5),ORIG(5),FACT(5)
DATA NAMESV/'EPHI','EZRI','BLOC'/
DATA NBITSV/6,6,3/
DATA NAMESH/'X   ','Y   ','Z   ','E   ','ELOS'/
DATA NBITSH/5*16/
DATA ORIG/3*1000.,0.,0./
DATA FACT/3*10.,2*100./

    CALL GSDET ('ECAL','BLOC',3,NAMESV,NBITSV,2,100,100,
   +                         ISET,IDET)
    CALL GSDETH('ECAL','BLOC',5,NAMESH,NBITSH,ORIG,FACT)
```

CALL GFDETH(IUSET,IUDET,NH*,NAMESH*,NBITSH*,ORIG*,FACT*)

Returns the hit parameters for detector IUDET of set IUSET. All arguments are explained above.

Author(s) : R. Brun
Origin : Same

Submitted: 01.11.83
Revised: 28.04.86

Handling Sensitive DETector Digitisation parameters

```
CALL GSDETD(IUSET,IUDET,ND,NAMESD,NBITSD)
```

Defines digitisation parameters for detector IUDET of set IUSET.

IUSET	user set identifier
IUDET	user detector identifier
ND	number of elements per digitisation
NAMESD	the ND variable names for the digitisation elements
NBITSD	the ND bit numbers for packing the variable values.

The routine is used at initialisation time once the geometrical volumes have been defined to describe the digitisation elements and the way to do packing in memory and on tape. Let us use the same example as in GSDETH. The non geometrical information we want to store for each digitisation is for example:

- ADC pulse height in a lead glass block

Example of one digitisation in that scheme:

```
EPHI 12
EZRI 41
BLOC  3
ADC  789
```

The FORTRAN coding to define the digitisation information could be:

```
DATA NAMESD/'ADC '/
DATA NBITSD/16/

      CALL GSDETD('ECAL','BLOC',1,NAMESD,NBITSD)
```

```
CALL GFDETD(IUSET,IUDET,ND*,NAMESD*,NBITSD*)
```

Returns the digitisation parameters for detector IUDET of set IUSET. All arguments as explained in GSDETD.

Author(s) : J.J.Dumont, W.Gebel†
Origin : Same

Submitted: 01.11.83
Revised: 01.10.84

Storing and Retrieving User Detector parameters

```
CALL GSDETU(IUSET,IUDET,NUPAR,UPAR)
```

Stores user parameters for detector IUDET of set IUSET.

IUSET	user set identifier
IUDET	user detector identifier
NUPAR	number of user parameters
UPAR	array of NUPAR user floating point parameters.

The routine is used at initialisation time once the geometrical volumes have been defined.

```
CALL GFDETU(IUSET,IUDET,NUPAR,NW*,UPAR*)
```

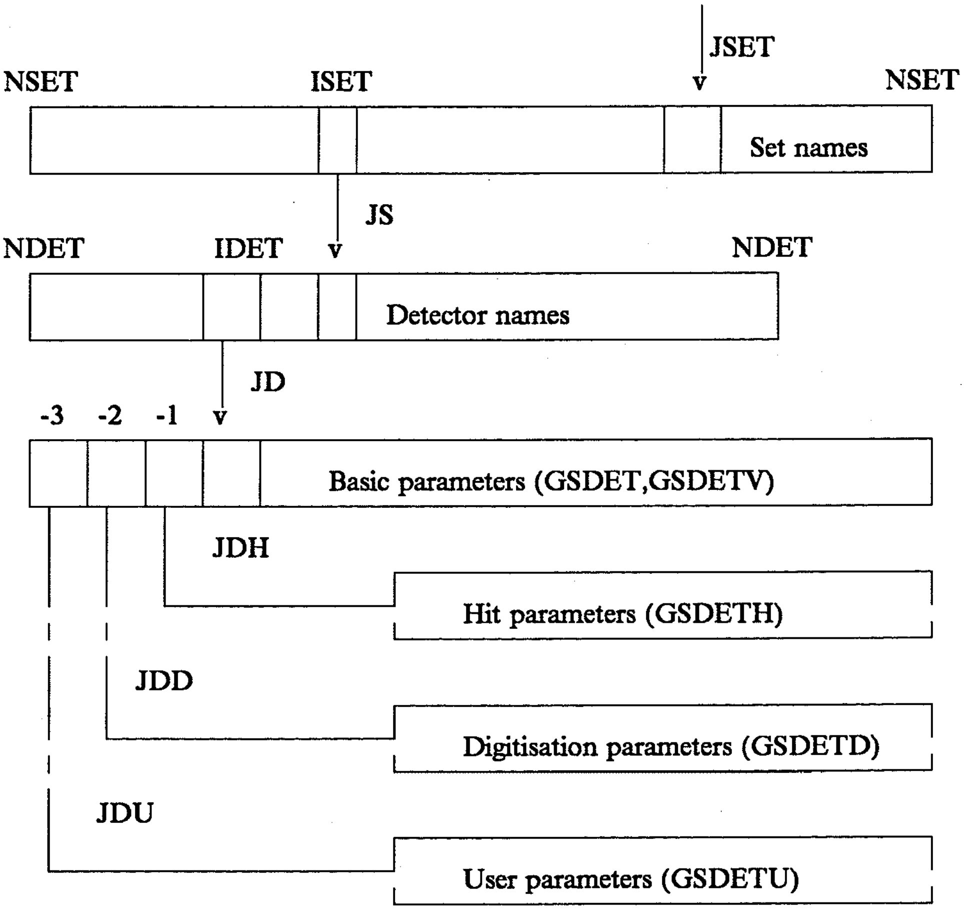
Returns the user parameters for detector IUDET of set IUSET.

IUSET	user set identifier
IUDET	user detector identifier
NUPAR	number of requested user parameters
NW	effective number of user parameters as stored in GSDETU
UPAR	array of NUPAR user floating point parameters.

Author(s) : R.Brun
Origin : Same

Submitted: 01.11.83
Revised: 21.08.87

The Detector Set data structure JSET



JS = LQ(JSET − ISET) pointer to detector set number ISET

The JSET data structure is filled by GSDET GSDETV(+GGDETV), GSDETH, GSDETD, GSDETU and eventually by GSDETA.

Author(s) : R.Brun, F.Bruyant, W.Gebel†
Origin : R.Brun, W.Gebel†, M.Maire

Submitted: 01.11.83
Revised: 21.08.87

Routines to Communicate with the JHITS data structure

CALL GSAHIT(ISET,IDET,ITRA,NUMBV,HITS,IHIT)*

Stores element values for current hit into the data structure JHITS.

ISET set number (can be obtained from /GCSETS/ filled by GFINDS)
IDET detector number (can be obtained from /GCSETS/ filled by GFINDS)
ITRA track number producing this hit
NUMBV array of volume numbers corresponding to list NAMESV of GSDET
HITS array of values for current hit elements
IHIT on return, current hit number. If = 0, hit has not been stored.

CALL GSCHIT(ISET,IDET,ITRA,NUMBV,HITS,NHSUM,IHIT)*

Same action as GSAHIT, but in case the physical volume specified by NUMBV contains already a hit for the same track then the routine will make a cumulative sum for the last NHSUM elements of the hit.

That facility is particularly interesting in case one want to store hits generated into a calorimeter.

CALL GPHITS(IUSET,IUDET)

Prints JHITS banks for detector IUDET of set IUSET.

IUSET user set identifier
if IUSET = 0 prints all JHITS banks of all sets
IUDET user detector identifier
if IUDET = 0 prints hits in all detectors of set IUSET.

```
CALL GFHITS(IUSET,IUDET,NVDIM,NHDIM,NHMAX,ITRS,NUMVS
         ,ITRA*,NUMBV*,HITS*,NHITS*)
```

Returns the hits produced by track ITRS (if 0, by all tracks) in the physical volume specified by the list NUMVS with generic volume name IUDET belonging to set IUSET.

IUSET	user set identifier
IUDET	user detector identifier (name of the corresponding sensitive volume)
NVDIM	1st dimension of arrays NUMBV and NUMVS (≤ argument NV of GSDET ≥ 1)
NHDIM	1st dimension of array HITS (≤ argument NH of GSDETH)
NHMAX	maximum number of hits to be returned
ITRS	number of the selected track. If ITRS = 0, all tracks are taken
NUMVS	is a 1-Dim array that must contain on input the list of volume numbers which identify the selected detector volume. Number 0 interpreted as 'all valid numbers'
ITRA	is a 1-Dim array that will contain on output for each hit the number of the track which has produced it
NUMBV	2-Dim array that will contain on output for each hit the list of volume numbers which identify each physical volume. Zeroed when no more volumes are stored
HITS	2-Dim array that will contain the NHITS hits
NHITS	returns the number of selected hits. In case the total number of hits is greater than NHMAX, NHITS is set to NHMAX + 1 and only NHMAX hits are returned

HITS(1,I)	is the element 1 for hit number I
NUMBV(1,I)	is the volume number 1 for hit number I
ITRA(I)	is the track number corresponding to hit number I

In the calling routine the arrays NUMVS, NUMBV, HITS and ITRA must be dimensioned to:

```
NUMVS(NVDIM)
NUMBV(NVDIM,NHMAX)
HITS(NHDIM,NHMAX)
ITRA(NHMAX)
```

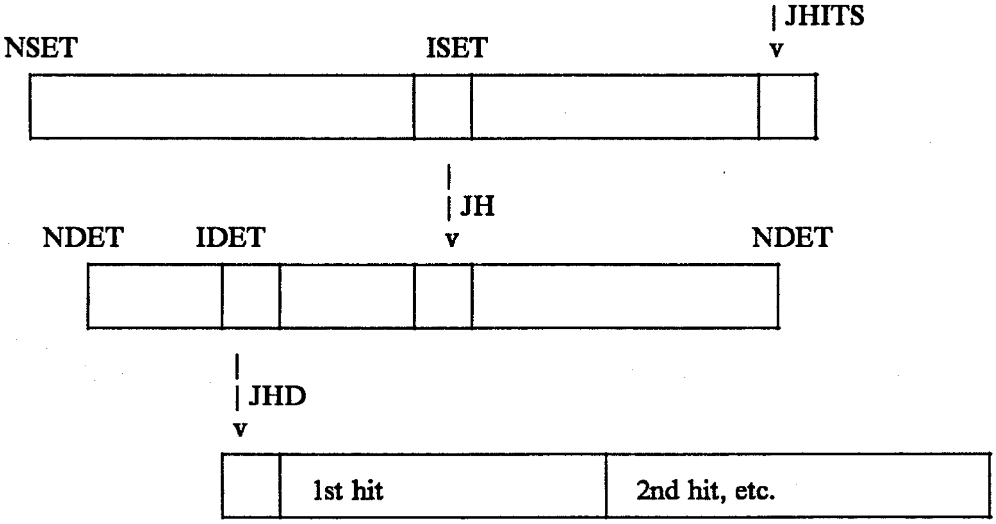
```
CALL GFPATH(ISET,IDET,NUMBV,NLEV,LNAM,LNUM)
```

Returns the list of NLEV volume names (LNAM) and numbers (LNUM) which identify the path through the JVOLUM data structure for the volume corresponding to the detector at position IDET in set at position ISET identified by the list of volume numbers NUMBV (see GFHITS).

Author(s) : R. Brun
Origin : Same

Submitted: 01.11.83
Revised: 28.04.86

The Hit data structure JHITS



Bank layout

JH = LQ(JHITS − ISET,) pointer to hits for set number ISET
JHD = LQ(JH − IDET), pointer to hits of detector IDET of set ISET
IQ(JH + IDET) number of words used so far for storing the hits of detector IDET
IQ(JHD + 1) 1st word of 1st hit
IQ(JHD + NWH + 1) 1st word of 2nd hit

JS = LQ(JSET − ISET)
JD = LQ(JS − IDET)
NWH = IQ(JD + 3)

The JHITS structure is filled with the routines GSAHIT and GSCHIT. The routine GFHITS can be used to get the hits for a detector IDET in set ISET.

Author(s) : W.Gebel†
Origin : R.Brun, W.Gebel†

Submitted: 10.08.84
Revised: 20.05.86

Routines to Communicate with the data structure JDIGI

```
CALL GSDIGI(ISET,IDET,LTRA,NTRA,NUMBV,KDIGI,IDIG*)
```

Stores element values for current digitisation into the data structure JDIGI.

ISET set number
IDET detector number
LTRA list of NTRA track numbers producing this digitisation
NUMBV volume numbers corresponding to list NAMESV of GSDET
KDIGI integer array of values for current digisation elements
IDIG on return, current digitisation number.
If = 0 digitisation has not been stored.

```
CALL GPDIGI(IUSET,IUDET)
```

Prints JDIGI banks for detector IUDET of set IUSET.

IUSET user set identifier
if IUSET = 0 prints all JDIGI banks of all sets
IUDET user detector identifier
if IUDET = 0 prints digitisations in all detectors of set IUSET

```
CALL GFDIGI(IUSET,IUDET,NTDIM,NVDIM,NDDIM,NDMAX,NUMVS,
                 LTRA*,NTRA*,NUMBV*,KDIGI*,NDIGI*)
```

Returns the digitisations for the physical volume specified by the list NUMVS with generic volume name IUDET belonging to set IUSET.

IUSET user set identifier
IUDET user detector identifier (name of the corresponding sensitive volume)
NTDIM 1st dimension of LTRA (max. number of tracks contributing)
NVDIM 1st dimension of NUMVS, NUMBV (argument NV of GSDET)
NDDIM 1st dimension of KDIGI (argument ND of GSDETD)
NDMAX is the maximum number of digitisations to be returned
NUMVS is a 1 − Dim array that must contain on input the geometric path of the detector volume to be selected.
All 0 interpreted as 'all physical volumes with generic name IUDET'
LTRA is a 2 − Dim array that will contain on output for each digitisation the numbers of the tracks which have produced it

NTRA is a 1−Dim array that will contain on output for each digitisation the total number of tracks contributing.
In case this number is greater than NTDIM, only the first NTDIM corresponding tracks can be returned on LTRA

NUMBV is a 2−Dim array that will contain on output for each digitisation the list of volume numbers which identify each physical volume

KDIGI is a 2−Dim array that will contain the NDIGI digitisations

NDIGI is the total number of digitisations in this detector.
In case the total number of digitisations is greater than NDMAX, NDIGI is set to NDMAX + 1 and only NDMAX digitisations are returned

- KDIGI(1,I) = digitisation type 1 for digitisation number I
- NUMBV(1,I) = volume number 1 for digitisation number I
- LTRA (1,I) = first track number contributing to digitisation number I

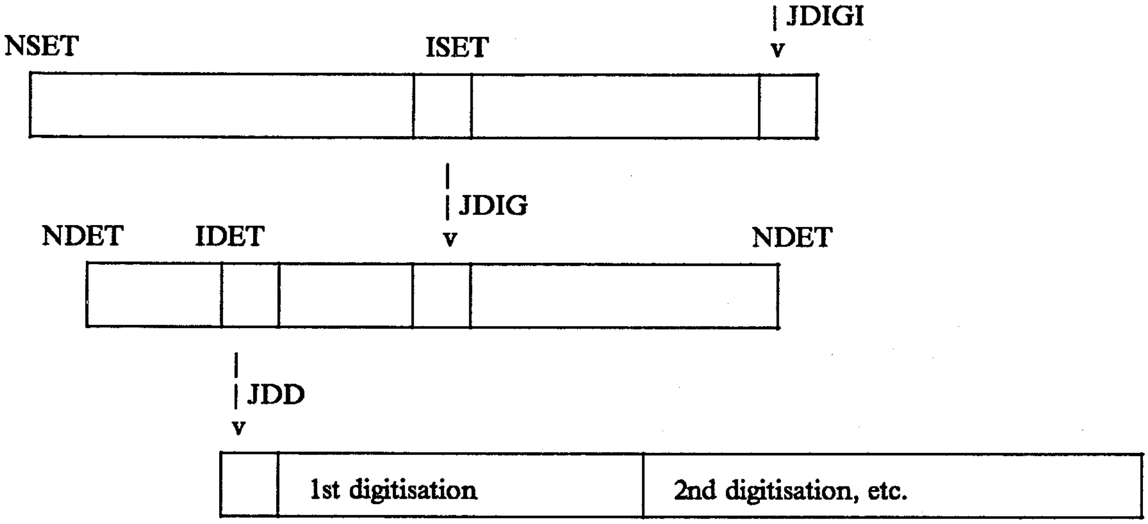
In the calling routine, the arrays LTRA, NTRA, NUMVS, NUMBV, KDIGI must be dimensioned to:

```
LTRA (NTDIM,NDMAX)
NTRA (NDMAX)
NUMVS(NVDIM)
NUMBV(NVDIM,NDMAX)
KDIGI(NDDIM,NDMAX)
```

Author(s) : R. Brun
Origin : Same

Submitted: 01.11.83
Revised: 28.04.86

The Digitisation data structure JDIGI



Bank layout

JDIG	= LQ(JDIGI − ISET), pointer to digitisations for set ISET
JDD	= LQ(JDIG − IDET), pointer to digitisations of detector IDET of set ISET
IQ(JDIG+IDET)	pointer to last word of last digitisation for this detector
IQ(JDD+1)	1st word of first digitisation
IQ(JDD+N+1)	1st word of second digitisation

JS = LQ(JSET − ISET)
JD = LQ(JS − IDET)
NWD = IQ(JD + 5)
NTRA number of tracks contributing to first digitisation
N = NWD + NTRA/2 + 1 (N varies from digitisation to digitisation)

The JDIGI structure is filled with the routine GSDIGI. The routine GFDIGI can be used to get the digitisations for a detector IDET in set ISET.

Author(s) : R.Brun, J.J.Dumont
Origin : GEANT2, H.Boerner

Submitted: 01.10.81
Revised: 21.05.86

Intersection of a Track with a Cylinder or a Plane

```
CALL GICYL(R,X1,X2,S1,S2,IC,XINT*,SINT*,PZINT*,IFLAG*)
```

Calculates intersection of track (x1,x2) with cylindrical detector of radius R. The track is approximated by a cubic in the track length. To improve stability, the coordinate system is shifted.

R	radius of cylinder in cm
X1	x,y,z,xp,yp,zp of 1st point
X2	x,y,z,xp,yp,zp of 2nd point
S1(2)	S at 1st (2nd) point
IC	= 1 straight line defined by x + xp
IC	= 2 straight line defined by x1 + x2
IC	= 3 cubic model
XINT	x,y,z,xp,yp,zp at intersection point
SINT	S at intersection point
PZINT	phi,z,dphi/dr,dz/dr
IFLAG	= 1 if track intersects cylinder, = 0 if not

```
CALL GIPLAN(YC,X1,X2,S1,S2,IC,XINT*,SINT*,PZINT*,IFLAG*)
```

Calculates intersection of track (x1,x2) with plane parallel to (X − Z). The track is approximated by a cubic in the track length. To improve stability, the coordinate system is shifted.

YC	Y coordinate of plane
X1,...	as for GICYL
IFLAG	= 1 if track intersects plane, = 0 if not

Warning: the default accuracy is 10 microns. The value of EPSI (internal variable) must be changed for a better precision.

Author(s) :
Origin : GEANT2

Submitted: 01.10.81
Revised: 28.05.86

Digitisation of Drift − or MWP − Chambers

```
CALL GPDRIF(DETREP,HITREP,IOUT*)
```

Digitisation routine for a plane drift chamber.

DETREP(1)	number of wires
DETREP(2)	wire spacing
DETREP(3)	SIN(alpha) (alpha = angle of the normal to the wire with respect to axis 1)
DETREP(4)	COS(alpha)
DETREP(5)	distance of wire 1 from the origin
DETREP(6)	drift velocity (cm/nsec)
HITREP(1)	x coordinate of intersection
HITREP(2)	y coordinate of intersection
IOUT(1)	wire number
IOUT(2)	drift time (signed to avoid left/right ambiguity)

```
CALL GCMWPC(DETREP,HITREP,IOUT*)
```

Routine to compute one or two digitisations produced by a hit on a cylindrical MWPC.

DETREP(1)	number of wires
DETREP(2)	wire spacing (radians)
DETREP(3)	dtheta/dz along the wires
DETREP(4)	theta of a point on wire 1
DETREP(5)	z of a point on wire 1
DETREP(6)	gap width
HITREP(1)	theta coordinate of intersection
HITREP(2)	z coordinate
HITREP(3)	dtheta/dr
HITREP(4)	dz/dr

IOUT(1)	wire number (− 1 − missing)
IOUT(2)	cluster size
IOUT(3)	wire number of second cluster if any
IOUT(4)	cluster size

CALL GPMWPC(DETREP,HITREP,IOUT*)

Digitisation routine for a plane MWPC.

DETREP(1)	number of wires
DETREP(2)	wire spacing
DETREP(3)	SIN(alpha) (alpha=angle of the normal to the wire with respect to axis 1)
DETREP(4)	COS(alpha)
DETREP(5)	distance of wire 1 from the origin
DETREP(6)	gap width
HITREP(1)	x coordinate of intersection
HITREP(2)	y coordinate
HITREP(3)	dx/dz
HITREP(4)	dy/dz

IOUT(1)	wire number
IOUT(2)	cluster size

Author(s) : W. Mitaroff
Origin : Same

Submitted: 21.02.85
Revised: 22.04.86

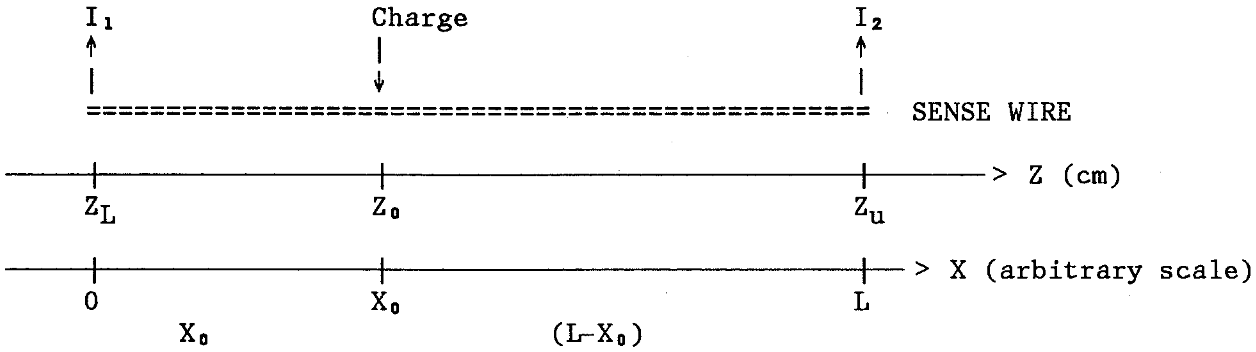
Digitisation of Drift Chambers

```
CALL GCDRIF(RADD,ZMIN,ZMAX,DETREP,HITREP,IOUT*)
```

Digitisation routine for a cylindrical drift chamber.

RADD	radius of cylinder in cm
ZMIN	z of lower end of cylinder
ZMAX	z of upper end of "
DETREP(1)	number of wires
DETREP(2)	wire spacing in PHI (radians)
DETREP(3)	cosine of wire angle
DETREP(4)	sine of wire angle (signed like dphi/dz)
DETREP(5)	dphi/dz along wire
DETREP(6)	phi of point with z=0 on wire 1
DETREP(7)	drift velocity (cm/nsec)
DETREP(8)	quantity describing the drift angle if.ne.0 = = > user routine GUDTIM
HITREP(1)	phi coordinate of intersection
HITREP(2)	z coordinate
HITREP(3)	dphi/dr
HITREP(4)	dz/dr
IOUT(1)	wire number (1..NWI with increasing phi) (−1 for bad DETREP parameters)
IOUT(2)	drift time (nsec) (+/− for phi(hit)>/< phi(wire)
IOUT(3)	digitised current division information (rel. pos. along wire of charge) (per mille)
IOUT(4)	amount of charge deposited to wire

Coordinate systems along wire



The scaling used is such that L = 1000.

Knowing the position Z_0 of the deposit of charge,

$$X_0 = L \bullet \frac{Z_0 - Z_L}{Z_u - Z_L}$$

This information is stored into IOUT(3).

CALL GCDERR(ICD,ERP,ERS)

Routine to calculate the error on the current division information as obtained by "GCDRIF".

ICD digitized current division information (0 ... 1000)
ERP variance of Gaussian distributed pedestral errors on the measured pulse heights relative to the sum of the pulse heights
ERS variance of Gaussian distributed slope errors on the measured pulse heights relative to the each pulse heights

Here we assume that X_0 has been determined by measuring the pulse heights I_1, I_2 with some statistical errors.

X_0 is then given by the formula

$$X_0 = L \bullet I_2/I_+ \quad \text{with} \quad I_+ = I_1 + I_2$$

and its error is determined by

$$\delta X_0 = -(X_0/I_+)\,\delta I_1 + (L - X_0/I_+)\,\delta I_2$$

with the errors on measuring the pulse heights

$$\delta I_1 = \delta_1 + \epsilon_1 \bullet I_1$$
$$\delta I_2 = \delta_2 + \epsilon_2 \bullet I_2$$

δ_1, δ_2 are of dimension (I) and represent the "pedestal" errors;
ϵ_1, ϵ_2 are the "slope" errors.
All are assumed to be distributed ***independently*** (no correlations), ***randomly*** and Gaussian around zero. This gives the final result

$$\delta X_0 = \underbrace{-\frac{\delta_1}{I_+} X_0 + \frac{\delta_2}{I_+} (L-X_0)}_{\text{"pedestal"}} + \underbrace{(\varepsilon_2-\varepsilon_1) \frac{X_0(L-X_0)}{L}}_{\text{"slope"}}$$

In GCDERR, the X_0 derived from GCDRIF is set to

$$X_0 = X_0 + \delta X_0 \quad (\text{but } 0 \leq X_0 \leq L)$$

using ERP ..... variance for δ_1/I_+ , δ_2/I_+ distributions

ERS ..... variance for ϵ_1, ϵ_2 distributions.

GEANT 3.11 USER'S GUIDE IOPA 001

Author(s) : R. Brun
Origin : GEANT2/3

Submitted: 15.08.84
Revised: 20.08.87

The I/O service routines

The I/O routines permit to read, write or print the GEANT3 data structures. When I/O with tapes is performed, the data structures can be handled in native mode or in machine independent format, using either the FZ package of ZEBRA (for GGET, GSAVE) or the RZ package (for GRGET, GRSAVE).

Events can be generated on one type of machine, for example full detector simulation, and the data produced can be analyzed on a different machine.

The data structures written to tape or disk can be read either in whole or in part.

Author(s) : R. Brun
Origin : GEANT3

Submitted: 01.06.83
Revised: 20.08.87

Open/Close FZ Zebra logical Units

CALL GOPEN(LUN,CHOPT,LEN,IER)*

LUN	Logical unit number
CHOPT	Character string or variable. If CHOPT = 'I' file is an input file 'O' file is an output file 'X' file is written in Exchange mode
LEN	Maximum record length. For more details see routine FZFILE in the ZEBRA manual
IER	Error flag

CALL GCLOSE(LUN,IER)*

LUN	Logical unit to be closed
IER	Error flag

Author(s) : R.Brun
Origin : GEANT2/3

Submitted: 01.06.83
Revised: 20.05.86

Read or Write Data Structures

CALL GGET(LUN,KEYS,NKEYS,IDENT,IER*)*

LUN	Logical unit number
KEYS	Keywords to select data structures
NKEYS	Number of keywords if negative, equivalent to 1 with KEYS(1)='INIT'
IDENT	Record identifier, 0 for INIT option
IER	Error flag

The following keywords can be specified:

PART to read the JPART data structure
MATE to read the JMATE (basic) data structure
TMED to read the JTMED data structure
VOLU to read the JVOLUM data structure
ROTM to read the JROTM data structure
SETS to read the JSET data structure
DRAW to read the JDRAW data structure

INIT to read all the above data structures

KINE to read the JVERTX,JKINE data structures
JXYZ to read the JXYZ (space points)
HITS to read the JHITS data structure
DIGI to read the JDIGI data structure

CALL GSAVE(LUN,KEYS,NKEYS,IDENT,IER)*

LUN	Logical unit number
KEYS	Keywords to select data structures
NKEYS	Number of keywords (Negative = INIT)
IDENT	Record identifier (0 for INIT)
IER	Error flag

The keywords mentioned above can be specified.

Author(s) : R. Brun
Origin : Same

Submitted: 20.08.87
Revised:

Open/Close RZ Zebra logical Units

CALL GRFILE(LUN,CHDIR,CHOPT)

LUN	Logical unit number
CHDIR	Name of the RZ top directory
CHOPT	Options For more details see routine FZFILE in the ZEBRA manual

CALL GREND(CHDIR)

CHDIR	Name of the RZ top directory (See RZEND in ZEBRA manual).

Author(s) : R. Brun
Origin : Same

Submitted: 20.08.87
Revised:

Read/Write RZ Data Structures

CALL GRGET(KEYSU,ID1,ID2,ID3,ICYCLE)

Read a data structure from an RZ file.

KEYSU	Keyword to select data structure(s)
ID1,2,3	First, second, third RZ key identifiers
ICYCLE	Cycle number.

In addition to the usual data structure names the following global keywords can be used.

```
KINE  -->  DRAW + MATE + PART + ROTM
         + RUNG + SETS + TMED + VOLU

KINE  -->  KINE + VERT

.TRIG  -->  DIGI + HEAD + HITS + KINE + VERT + JXYZ
```

CALL GRSAVE(KEYSU,ID1,ID2,ID3,ICYCLE)

Write a data structure into an RZ file.

Arguments as in GRGET.

Author(s) : F.Bruyant
Origin : GEANT2/3

Submitted: 28.04.86
Revised:

Introduction to the section KINE

This section describes the GEANT3 kinematics structures JVERTX and JKINE and the routines which permit to store and retrieve the relevant information : GSVERT, GPVERT, GSKINE, GFKINE, GPKINE.
It also contains a short description of the GEANT3 interface to the Lund generator and related routines.

Author(s) : R. Brun
Origin : GEANT2

Submitted: 01.06.83
Revised: 21.08.87

Storing/Retrieving Vertex and Track parameters

```
CALL GSVERT(VERT,NTBEAM,NTTARG,UBUF,NUBUF,NVTX*)
```

Stores vertex parameters.

VERT array of (x,y,z) position of the vertex
NTBEAM beam track number origin of the vertex
= 0 if none exists
NTTARG target track number origin of the vertex
UBUF user array of NUBUF floating point numbers
NUBUF
NVTX new vertex number (= 0 in case of error).

```
CALL GPVERT(IVTX)
```

Prints vertex parameters.

IVTX for vertex IVTX.
(for all vertices if IVTX = 0)

```
CALL GSKINE(PLAB,IPART,NV,UBUF,NUBUF,NT*)
```

Stores long life track parameters.

PLAB components of momentum
IPART particle number (see GSPART)
NV vertex number origin of track
UBUF array of NUBUF floating point user parameters
NUBUF
NT track number (if = 0 error).

CALL GFKINE(ITRA,VERT,PVERT*,IPART*,NVERT*,UBUF*,NUBUF*)*

Retrieves long life track parameters.

ITRA	track number for which parameters are requested
VERT	vector origin of the track
PVERT	4 momentum components at the track origin
IPART	particle type (= 0 if track ITRA does not exist)
NVERT	vertex number origin of the track
UBUF	user words stored in GSKINE.

CALL GPKINE(ITRA)

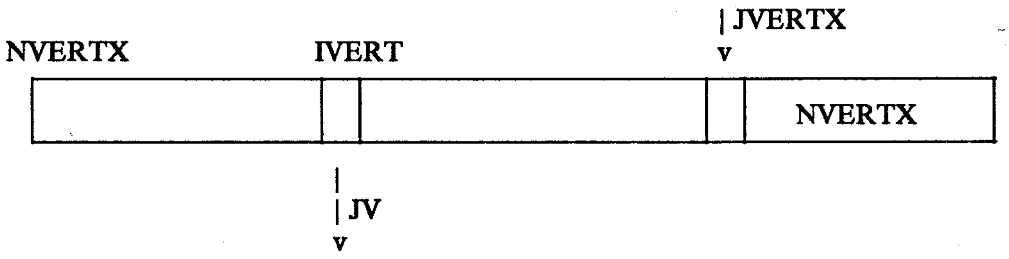
Prints initial track parameters.

ITRA	for track ITRA (for all tracks if ITRA = 0)

Author(s) :
Origin : GEANT2/3

Submitted: 01.11.83
Revised: 28.05.86

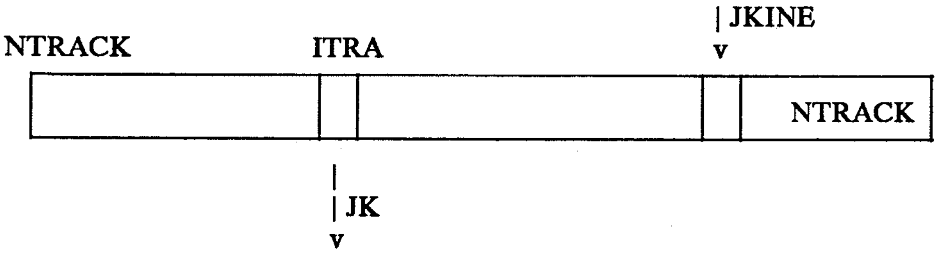
The Kinematics data structures JVERTX and JKINE



1	VX
2	VY
3	VZ
4	Time of flight
5	NTBEAM
6	NTTARG
7	no. of tracks
8	List of tracks
9	...

JV = LQ(JVERTX − IVERT) pointer to parameters of vertex number IVERT
Q(JV+1) x coordinate of this vertex.

The JVERTX banks are filled by the routine GSVERT.



1	PX
2	PY
3	PZ
4	Energy
5	IPART
6	Vertex origin
7	no. of vertices
8	List of vertices
9	...

JK	= LQ(JKINE − ITRA) pointer to parameters of track number ITRA
Q(JK+1)	x component of the momentum for this track.

The JKINE banks are filled by the routine GSKINE.

The track parameters may be obtained through the routine GFKINE.

Author(s) : R.Hemingway
Origin : R.Brun,R.Hemingway

Submitted: 01.11.83
Revised: 28.05.86

GEANT3 Interface to the Lund Monte-Carlo

CALL GLUND

GEANT3 is now interfaced with the Lund Monte-Carlo JETSET 6.2 through the routine GLUND which may be called from GUKINE.

For technical details of the Lund Monte-Carlo, see the original reports

1) The Lund Monte-Carlo for Jet fragmentation - T. Sjostrand
LU/TP/82-3 and Computer Physics Communication 27 (1982) 243

2) The Lund Monte-Carlo for e^+ e^- Jet Physics - T.Sjostrand
LU/TP/82–7 and Computer Physics Communication 28 (1983) 229

and the CERN Program Library Long Write-up W5035.

The routine GLUND calls the routine GLUNDI which sets up the following default parameters for event generation :

MSTE(1) = 2 to allow QQ,QQG,QQGG and QQQQ jets
MSTE(2) = 2 to generate with QFD and SU(2)*U(1)
MSTE(4) = 6 to allow up to 6 flavours U,D,S,C,B,T
MSTE(7) = 1 to add the initial state radiative corrections
PMAS(106) = 30. to set the top quark mass at 30 GeV
IDB(37,43,45,46,47,57,70) = 0 to inhibit the decay of K0S, Σ^+ , Σ^-, Ξ^0, Ξ^-, λ, Ω^-
IFLUND = 0 to select the flavour code from 1 to 6 (0 = mixture)
ECLUND = 94. to enter the total CMS energy.

The routine GLUNDI may also be called by the user in UGINIT before the call to GFFGO, in which case the default parameters can be changed through the data cards MSTE, KTYP, PMAS, PWID, IDB and LUND [BASE 040].

Author(s) : R.Hemingway
Origin : Same

Submitted: 01.11.83
Revised: 28.05.86

TAU+ TAU− Generation and Decay

CALL GTAU

The process $e^+ e^-$ −> Tau+ Tau− can be generated through the routine GTAU which may be called from GUKINE.
The total CMS energy is assumed to be 94 GeV. and the Tau mass 1.7842 GeV.

The Tau decay can be performed through the routine GLUDKY which is interfaced with the Lund Monte-Carlo and should be called from the user routine GUDCAY.

Author(s) : M. Maire
Origin : GEANT3

Submitted: 24.09.84
Revised: 26.05.86

Introduction to the section PHYS

1. Summary of the Physics Processes

In the computer simulation of the tracking of particles through an experimental setup, one has to take into account their interactions with matter. GEANT considers the dominant processes which can occur in the **energy range from 10KeV up to 10TeV.**
Simulating a given process means:

- Evaluate the probability of occurence of the process, by sampling the **total cross-section.**
- Generate the final state after interaction, by sampling the **differential cross-section** of the process.
- Compute the mean value of some quantities which characterize the (quasi) continuous processes (energy losses, multiple scattering)

In the Table below we summarize all the processes currently implemented in GEANT3, with a reference to the corresponding routines.

	Computation of total cross-section and/or energy losses	Generation of the final state particles
Processes involving the Photon		
(e^+ , e^-) pair conversion	PHYS 210	PHYS 211
Compton collision	PHYS 220	PHYS 221
Photo electric effect	PHYS 230	PHYS 231
Photo fission of heavy elements	PHYS 240	PHYS 240
Processes involving e^-/e^+		
Multiple scattering		PHYS 320 **or** 325
Ionisation and delta rays production	PHYS 330	PHYS 331 **or** 332
Bremsstrahlung	PHYS 340	PHYS 341
Annihilation of positron	PHYS 350	PHYS 351
Processes involving μ^-/μ^+		
Decay in flight	CONS 310	PHYS 400
Multiple scattering		PHYS 320 **or** 325
Ionisation and delta rays production	PHYS 430	PHYS 331 **or** 332
Bremsstrahlung	PHYS 440	PHYS 441
Direct (e^+ ,e^-) pair production	PHYS 450	PHYS 451
Nuclear interaction	PHYS 460	PHYS 460
Processes involving Hadrons		
Decay in flight	CONS 310	PHYS 400
Multiple scattering		PHYS 320 **or** 325
Ionisation and delta rays production	PHYS 430	PHYS 331 **or** 332
Hadronic interactions	PHYS 500 **or** 510	PHYS 500 **or** 510

2. General Comments

Hadronic Interactions. Two alternatives are proposed:

1) an old version ot TATINA package (from Baroncelli) is a part of the GEANT Pam file [PHYS 500].
The total cross-sections have been updated by R. Barlow (Manchester) in January 1984.

2) an interface with GHEISHA 7.03 package (from H.C. Fesefeldt) has been written by F. Carminati (CERN – DD) [PHYS 510].

The TATINA package has been kept for backward compatibility and fast checking and debugging. However, careful simulations should use the GHEISHA7 interface.

Electromagnetic Processes. By mean of systematic fits to the existing data, the cross-sections of the electromagnetic processes are well reproduced (within few percents) from 10KeV up to 100GeV, and for low Z as well for heavy materials.

This feature, together with the use of the GHEISHA interface, should make GEANT useful for careful shower simulations even in a gas.

The Muon interactions have been taken into account up to 10TeV, making GEANT available for cosmic rays studies.

Ionisation of the Matter by Charged Particles. Two alternatives are proposed:

1) Sample a Landau distribution around a mean value of the energy loss [PHYS 332]

2) Generate explicitly the delta rays [PHYS 330, 331, 430]

The 2 methods **cannot** be used at the same time (double counting). An automatic protection is put within GEANT. See PHYS 333, 332 and BASE 040 for further information.

Multiple Scattering. Two methods are proposed:

a Gaussian approximation [PHYS 320], or the Molière theory [PHYS 325]. By default GEANT uses the Gaussian approximation.

The JMATE data structure. In order to save time, the total cross-section and the DE/DX of all processes involving gamma, electron and muon are computed only once and tabulated as a function of the material and of the energy of the incident particle. Then one makes a linear interpolation through these tables when using it. See PHYS 100 and CONS 199 for a deeper understanding of these tables.

Probability of Interaction. The total cross-section of each process is used at tracking time to evaluate the probability of occurence of the process. More exactly to evaluate the distance to the point where the interaction will occur. See PHYS 010 for the explanation of the method used.

Note: The section PHYS is closely related to the section CONS. The user which would like to have a complete overview of the physics processes included in GEANT3 should read both sections.

3. Control of the physical processes

For most of thc individual proccsscs thc dcfault option (indicatcd by * in thc Tablc bclow) can bc changed via data cards [BASE 030, 040].

Below are listed the data card keywords, the flag names and values, and the resulting action.

Keyword	Flag name in GCPHYS	Flag=0	Flag=1	Flag=2
DCAY	IDCAY /GCPHYS/	NO decay in flight	Secondaries gen. part. stopped * ISTOP=1	part. stopped no secondaries
MULS	IMUL	NO multiple scattering	Gaussian scattering *	Molière scat.
PFIS	IPFIS	NO Photo fission *	part. stopped ISTOP=1 secondaries gen.	part. stopped ISTOP=2 no secondaries
MUNU	IMUNU	NO Nuclear Interaction *	μ not stopped secondaries gen.	μ not stopped no secondaries
LOSS	ILOSS	NO energy loss	from table	Landau fluctuat. *
PHOT	IPHOT		γ stopped ISTOP=1	γ stopped no e^- generated ISTOP=2 *
COMP	ICOMP		γ NOT stopped ISTOP=0	γ NOT stopped no e^- generat. ISTOP=0 *
PAIR	IPAIR		γ stopped ISTOP=1	γ stopped NO pair generat. ISTOP=2 *
BREM	IBREM		e^- NOT stopped ISTOP=0	e^- NOT stopped no γ generated ISTOP=0 *
DRAY	IDRAY		part. NOT stopp. ISTOP=0	part. NOT stopped no secondary gen. ISTOP=0 *
ANNI	IANNI		e^+ stopped ISTOP=1	e^+ stopped no γ generat. ISTOP=2 *
HADR	IHADR		hadron stopped ISTOP=1	hadron stopped no secondaries gen. ISTOP=2 *

Author(s) : M.Maire
Origin : GEANT2

Submitted: 20.12.84
Revised: 28.05.86

Compute the Occurence of a Process

Common block GCPHYS

1. Principle

For the complete simulation of a given process (Brems, Delta-rays, etc) several tasks have to be performed.

1) At the begining of the track, ***evaluate*** (i.e.: ***sample***) the probability that the particle interacts (routine GTSET).

2) During the tracking, ***survey*** the total length of the particle and ***update*** the probability for the occurence of the process. The corresponding code has been inserted directly within the tracking routines: GTELEC, GTGAMA, GTMUON, GTHADR and GTNEUT.

3) When the interaction occurs, ***generate*** the final state particles. This is generally done by specialized routines: GPHOT, GCOMP, GBREM, etc.

4) If the incident particle survives after the interaction, ***sample again*** the probability for a new interaction (i.e.: go to step 1). This is done at the end of the above specialized routines.
This happens for: Compton effect, Delta-rays production, Bremsstrahlung, Direct ($e^+ e^-$) production by muon, Nuclear interaction by muon.

It should be outlined that the evaluation of the step length is made ***independently for each process which can occur,*** the final step size being the ***minimum*** of all of them, and the process finally considered being the corresponding one.

2. Practical Method

The Mean Free path of particle for a given process, λ, is dependent of the medium and cannot be used directly to sample the probability of interaction, whereas the number of Mean Free Paths traversed,

$$N_\lambda = \int_{x_0}^{x} [dx/\lambda(x)] \qquad [1]$$

is not.

If N_R is a random variable denoting the number of mean free paths from a given point until the point of interaction, it can be shown that N_R has the distribution

$$P(N_R < N_\lambda) = 1-\exp(-N_\lambda) \qquad [2]$$

Therefore, the total number of mean free paths to travel up to the interaction point, N_λ, is sampled at the begining of the trajectory as:

$$N_\lambda = -\ln(R) \qquad [3]$$

where R is a random number in the range (0,1)

Then N_λ is permanently updated along the trajectory, according equation [1]:

$$N_\lambda = N_\lambda - \frac{\Delta x}{\lambda(x)} \qquad [4]$$

until $N_\lambda = 0$ which triggers the interaction.

3. COMMON GCPHYS

All the relevant informations are located within COMMON /GCPHYS/ which is shown below:

```
+SEQ,GCPHYS
```

```
      COMMON/GCPHYS/IPAIR,SPAIR,SLPAIR,ZINTPA,STEPPA
     +             ,ICOMP,SCOMP,SLCOMP,ZINTCO,STEPCO
     +             ,IPHOT,SPHOT,SLPHOT,ZINTPH,STEPPH
     +             ,IPFIS,SPFIS,SLPFIS,ZINTPF,STEPPF
     +             ,IDRAY,SDRAY,SLDRAY,ZINTDR,STEPDR
     +             ,IANNI,SANNI,SLANNI,ZINTAN,STEPAN
     +             ,IBREM,SBREM,SLBREM,ZINTBR,STEPBR
     +             ,IHADR,SHADR,SLHADR,ZINTHA,STEPHA
     +             ,IMUNU,SMUNU,SLMUNU,ZINTMU,STEPMU
     +             ,IDCAY,SDCAY,SLIFE ,SUMLIF,DPHYS1
     +             ,ILOSS,SLOSS,SOLOSS,STLOSS,DPHYS2
     +             ,IMULS,SMULS,SOMULS,STMULS,DPHYS3
```

The 9 first processes (from PAIR production up to MUon NUclear interaction) have the same scheme. So let consider just the first one. See [BASE 030] for a complete description of the COMMON.

SLPAIR last point along the trajectory where N_λ has been updated according [4].
Usually it is the begining of the current step
SLPAIR = SLENG−STEP

ZINTPA N_λ = remaining number of mean free paths [4] evaluated at SLPAIR
STEPPA $\lambda(x)$ = value of the mean free path in the current medium (i.e.: at SLPAIR)
SPAIR $N_\lambda{*}\lambda$ = potential remaining step length before interaction, evaluated at SLPAIR
IPAIR flag for secondaries:
1 = generation of secondaries
2 = no generation of secondaries

The Decay in flight is simpler since the mean life time of the particle (τ) is not material dependant and can be sampled directly.

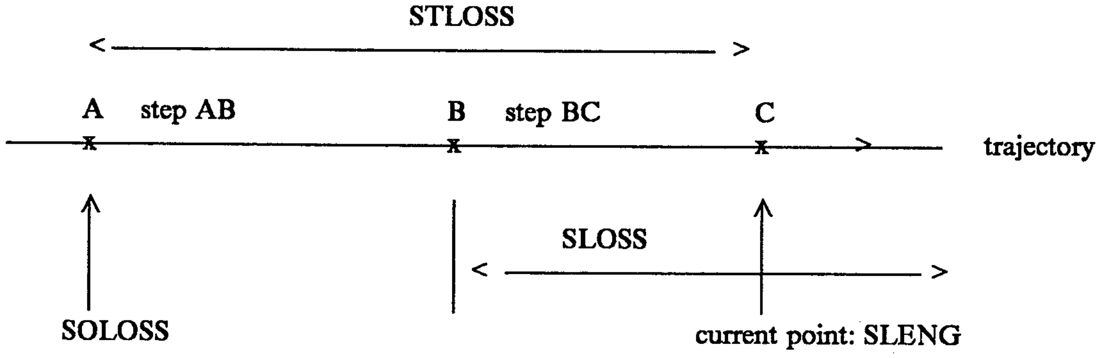
SLIFE $-c\tau\ln(R)$ = total proper life time, sampled at the creation of the particle
SUMLIF elapsed proper time since the creation of the particle
SDCAY distance left to decay point based on current momentum value.

4. Continuous Energy Loss and Multiple Scattering

For efficiency reason, the continuous energy loss of the particle along its trajectory and the multiple scattering are not computed at each step.

The Figure below shows a case where the energy loss is computed in point A and C but not in B.

- In A the energy loss is computed
- the step AB is limited by nothing else than the multiple scattering or the maximum curvature in the magnetic field: i.e. there is no discrete physical process happening in B, nor boundary reached. In that case the energy lost is not computed at point B.
- the step BC is limited by some physical process, or because the particle reaches a boundary. Then the energy loss is also computed.
  At the end of step BC we have the following situation:



SOLOSS total track length at last point where the energy loss was computed
SLOSS potential step for the computation of the energy loss, evaluated in B
STLOSS total step length over which the energy loss has been computed.
Hence, in C, the mean value of the energy loss is:

$$\text{DEMEAN} = \frac{\text{DE}}{\text{DX}}(\text{B}) * \text{STLOSS} \qquad [5]$$

In addition, if ILOSS ≥ 2, a Landau fluctuation (DELAND) is applied around this mean value, and the energy lost from A to C is:

$$\text{DESTEP} = \text{DEMEAN} + \text{DELAND} \qquad [6]$$

The same scheme applies to multiple scattering

here IMULS = 1 : Gaussian approximation
IMULS = 2 : Molière theory.

Author(s) : R. Brun
Origin : Same

Submitted: 28.05.86
Revised:

Steering routine for Cross-sections and Energy loss calculation

```
CALL GPHYSI
```

The routine GPHYSI is called at initialisation time from a user routine like UGINIT (see example in BASE 100). The purpose of GPHYSI is twofold:

- print the parameters which will define the current run
- compute the cross-section and energy loss tables and fill the JMATE data structure [CONS 199].

1. Print the run definitions

Get from the JRUNG data structure [BASE 299] the GEANT version number and the ZEBRA version number.

Get from the JTMED data structure the tracking and physics parameters which will be valid for the current run. See the description of the JTMED data structure in [CONS 210] and [CONS 299].

A summary table like below will be outputed by GPHYSI.

```
****************************************************************
*                                                              *
*    G E A N T  Version 3.1061      DATE/TIME 860521/1208      *
*                                                              *
*                       R U N     1                            *
*                                                              *
****************************************************************
*                                                              *
*      Data structure   Date    Time     GVERSN    ZVERSN      *
*      --------------   ----    ----     ------    ------      *
*                                                              *
*           INIT       860521   1208     3.1061     3.41       *
*                                                              *
*           KINE       860521   1208     3.1061     3.41       *
*                                                              *
*           HITS       860521   1208     3.1061     3.41       *
*                                                              *
*           DIGI       860521   1208     3.1061     3.41       *
*                                                              *
*--------------------------------------------------------------*
*                                                              *
*              Standard TPAR for this run are                  *
*              ------------------------------                  *
*                                                              *
*  CUTGAM=500.00 KEV  CUTELE=500.00 KEV  CUTHAD=  1.00 MEV     *
*  CUTNEU=  1.00 MEV  CUTMUO=  1.00 MEV                        *
*  BCUTE =500.00 KEV  BCUTM =500.00 KEV                        *
*  DCUTE = 10.00 TEV  DCUTM = 10.00 TEV  PPCUTM= 10.00 MEV     *
*  IPAIR =        1.  ICOMP =        1.  IPHOT =        1.     *
*  IPFIS =        0.  IDRAY =        0.  IANNI =        1.     *
*  IBREM =        1.  IHADR =        2.  IMUNU =        0.     *
*  IDCAY =        1.  ILOSS =        2.  IMULS =        1.     *
*                                                              *
*     Special TPAR for TMED   2   LIQ NE213                    *
*     -------------------------                                *
*  CUTGAM= 50.00 KEV  CUTELE= 50.00 KEV  CUTHAD=  1.00 MEV     *
*  CUTNEU=  1.00 MEV  CUTMUO=  1.00 MEV                        *
*  BCUTE = 50.00 KEV  BCUTM = 50.00 KEV                        *
*  DCUTE = 10.00 TEV  DCUTM = 10.00 MEV  PPCUTM= 10.00 MEV     *
*  IPAIR =        1.  ICOMP =        1.  IPHOT =        1.     *
*  IPFIS =        0.  IDRAY =        0.  IANNI =        1.     *
*  IBREM =        1.  IHADR =        2.  IMUNU =        0.     *
*  IDCAY =        1.  ILOSS =        2.  IMULS =        1.     *
*                                                              *
*                                                              *
****************************************************************
```

The meaning of the parameters can be found in the description of the commons: /GCCUTS/ and /GCPHYS/ [BASE 030].

It is important to understand in which ways the default values of the parameters can be overwritten (see example in BASE 100):

1) a set of global default values for the parameters are defined within the routine GINIT. 'Global' means: default for all tracking medium.

2) These default values can be changed by data cards via the routine GFFGO. A summary of all valid data cards is given in [BASE 040].

3) After having set-up its tracking media and materials, the user can redefine again the tracking and physics parameters for **each medium** via the routine GSTPAR [CONS 210].

4) Alternately, if the data structures are read from an external file ('GET' data card) all the parameters are taken from this file.

2. Compute cross-section and energy loss tables

GPHYSI is the steering routine to compute the cross-section and energy loss tables for all materials effectively used as tracking medium.

GPHYSI builds and fills the JMATE data structure as described in CONS 199. Here, we give the flow chart of the calculations. The description of the specialized routines can be found in the rest of the section PHYS.

Note: if several tracking medium are using the same material (for instance a calorimeter and a chamber support can be both in steel) the cut parameters must be the same. If this is not possible, the user must define different materials.

GPHYSI

- GCOMPI Total cross-section of compton collision
- GPHOTI Total cross-section of photo electric effect
- GPFISI Total cross-section of photo fission
- GANNII Total cross-section of e^+ annihilation
- GMUNUI Total cross-section of Muon Nuclear interaction
- GBRELA Energy loss due to Bremsstrahlung:
  - GBRELE for electrons
  - GBRELM for muons
- GBRSGA Total cross-section of Bremsstrahlung
  - GBRSGE for electrons
  - GBRSGM for muons
- GDRELA Energy loss due to Ionisation:
  - GDRELE for electrons
  - GDRELP for protons
- GDRSGA Total cross-section of Delta rays production
- GPRELA Energy loss due to (e^+ ,e^-) production:
  - GPRELM by muons
- GPRSGA Total cross-section for (e^+ ,e^-) production:
  - GPRSGG by gamma
  - GPRSGM by muons

Notes

1) The Muon Nuclear interactions are treated **either** as a continuous energy lost by the muon (IMUNU=0) **or** as a discrete process (IMUNU≥ 1), **in an exclusive way**, i.e.:

   a) If IMUNU=0 , the DE/DX due to the interactions is computed together with that coming from direct (e^+ ,e^-) pair production in the routine GPRELA.

   b) If IMUNU≥ 1 , the total cross-section is computed in GMUNUI.

2) The total cross-sections for hadronic interactions cannot be tabulated at initialisation time, as they are too much dependant of the nature of the projectile. They are computed at tracking time by the FUNCTIONS:
   GHSIGM in case of TATINA [PHYS 500]
   GHESIG in case of GHEISHA [PHYS 510]

Author(s) : L.Urban
Origin : GEANT2

Submitted: 10.11.84
Revised: 30.05.86

Total cross-section for e^+ e^- production by Photons

1. Method

We have parametrized the total cross-section of (e^+ ,e^-) production by photon as:

$$\sigma(Z,E_\gamma) = Z(Z+1) \bullet (F_1(X)+F_2(X)Z+F_3(X)/Z) \quad \text{(barn/atom)} \qquad [1]$$

where $X = \ln(E_\gamma/m)$
m = electron mass
E_γ = photon energy

and $$F_i(X) = \sum_{n=0}^{n=5} c_n X^n$$

The parameters c_n were taken for a least square fit to the data (Ref.1) and can be found in DATA within the function GPRSGG(Z,E) which computes the formula [1] (in barn/atom)

This parametrisation gives a good description of the data in the range:

$$1 \leq Z \leq 100$$

$$1.5\text{MeV} \leq E_\gamma \leq 100\text{GeV}$$

$$\frac{\Delta\sigma}{\sigma} \leq 5\% \quad \text{with a mean value of} \simeq 2.2\%$$

2. Subroutine GPRSGA

The mean free path, λ, for a photon to interact via pair production is given by

$$\lambda = 1/\Sigma \qquad [2]$$

where Σ is the macroscopic cross-section (1/cm). This quantity is given by

$$\Sigma = \frac{N\rho\sigma(Z,E_\gamma)}{A} \qquad [3a]$$

in the case of a compound or mixture:

$$\Sigma = \frac{N\rho\sum_i p_i\sigma(Z_i,E_\gamma)}{\sum_i p_i A_i} = N\rho \sum_i (w_i/A_i) \bullet \sigma(Z_i,E_\gamma) \qquad [3b]$$

N Avogadro's number

$Z(Z_i)$ atomic number of the material (i-th component of the material)

$A(A_i)$ atomic weight of the material (i-th component)

ρ density of the material

σ total cross-section per atom for pair production

p_i proportion by number of the i-th element in the material $p_i \simeq w_i/A_i$ where w_i is the corresponding proportion by weight

This mean free path is tabulated at initialisation time as a function of the medium and of the energy by the routine GPRSGA. The energy binning is set within the array ELOW (common GCMULO) in the routine GPHYSI.

Pointers

JMA = LQ(JMATE − I) pointer to the I-th material

JPAIR = LQ(JMA − 10) pointer to pair production cross-section

3. References

1) J.H. Hubbell, H.A. Gimm, I. Overbo, Jou. Phys. Chem. Ref. Data *9*, 1023 (1980).

Author(s) : G.N.Patrick, L.Urban
Origin : GEANT2

Submitted: 12.12.84
Revised: 13.05.86

Simulation of e^+ e^- Pair production by Photons

CALL GPAIRG

GPAIRG generates photon pair production by using a modified version of the random number techniques of Butcher and Messel (see Ref. 1) to sample the secondary electron/positron energies from the Coulomb corrected Bethe-Heitler differential cross-section.

Input: via COMMON/GCTRAK/
Output: via COMMON/GCKING/

GPAIRG is automatically called, if, and when, the parent photon reaches its decay point during the tracking stage of GEANT.

1. Method

We give a very short summary of the random number technique used here (see [1], [2]). The method is a combination of the composition and rejection Monte Carlo methods.

Suppose we wish to sample x from the distrbution f(x) and the (normalized) probably density function can be written as

$$f(x) = \sum_{i=1}^{n} \alpha_i f_i(x) g_i(x) \qquad [1]$$

where, $\alpha_i > 0$
$f_i(x)$ density functions (normalised correctly)
$0 \leq g_i(x) \leq 1$

According to this method x can be chosen by

1) selecting a random integer i ($1 \leq i\ n$) with probability of selecting i being proportional to α_i

2) selecting a value x′ from the distribution $f_i(x)$

3) calculating $g_i(x')$ and accepting $x = x'$ with selection probability of $g_i(x')$ (if x′ is rejected we have to start again from step 1).

It can be shown that this scheme is a correct one and the mean number of tries to accept a value is

$$\sum_i \alpha_i$$

In practice we have a good method of sampling from the distribution $f(x)$ if

- all thc subdistributions $f_i(x)$ can be sampled easily
- the rejection functions $g_i(x)$ can be evaluated easily/quickly
- the mean number of tries is not too large.

Thus the different possible decompositions of the distribution $f(x)$ are not equivalent from the point of view of the practice (e.g. they can be very different in computational speed) and it can be very useful to make some kind of optimization.

A remark of pratical importance: if our distribution is not normalized ($\int f(x)dx = C > 0;\ C \simeq 1$), the method can be used in the same manner, the only difference is that the mean number of tries in this case is given by

$$\sum_1 \alpha_i / C$$

The Coulomb corrected Bethe-Heitler differential cross section for a photon of energy E to produce an electron pair one of which has an energy ϵE is given as [3]:

$$\frac{d\sigma(Z,E,\epsilon)}{d\epsilon} = \frac{r_0{}^2\alpha Z^2[Z+\xi(Z)]}{E^2}\{[\epsilon^2+(1-\epsilon)^2][\Phi_1(\delta)-F(Z)] + {}^2/_3\epsilon(1-\epsilon)[\Phi_2(\delta)-F(Z)]\} \qquad [2]$$

where, $\Phi_i(\delta)$ are the screening functions depending on the screening variable δ

$$\delta = \frac{136m}{Z^1/_3E}\ \frac{1}{\epsilon(1-\epsilon)} \qquad [3]$$

$$\left.\begin{array}{l} \Phi_1(\delta) = 20.867 - 3.242\delta + 0.625\delta^2 \\ \Phi_2(\delta) = 20.209 - 1.930\delta - 0.086\delta^2 \end{array}\right\} \delta \leq 1$$

$$\Phi_1(\delta) = \Phi_2(\delta) = 21.12 - 4.184\ln(\delta+0.952) \qquad \delta > 1 \qquad [4]$$

$$F(Z) = \begin{cases} {}^4/_3\ln Z & E < 0.05\mathrm{GeV} \\ {}^4/_3\ln Z + 4f_c(Z) & E \geq 0.05\mathrm{GeV} \end{cases} \qquad [5]$$

$$\xi(Z) = \frac{\ln(1440/Z^2/^3)}{\ln(183/Z^1/^3) - f_c(Z)},$$

$f_c(Z)$ the Coulomb correction function

$f_c(Z) = a{*}(1/(1+a)+0.20206-0.0369{*}a+0.0083{*}a^2-0.002{*}a^3)$
$a = (\alpha{*}Z)^2$
$x = E/E0 \quad E0 = 0.05\mathrm{GeV}$
$y = \ln(x)$

The kinematical range for the variable ϵ is

$$m/E \leq \epsilon \leq 1 - m/E \qquad [6]$$

The cross-section is symmetric with respect to the interchange of ϵ with $1-\epsilon$, so we can restrict ϵ to lie in the range

$$\epsilon_0 = m/E \leq \epsilon \leq 1/2$$

After some algebraic manipulations we can decompose the cross section as (note: the normalisation is not important):

$$\frac{d\sigma}{d\epsilon} = \sum_{i=1}^{2} \alpha_i f_i(\epsilon) g_i(\epsilon) \qquad [7]$$

where, $\alpha_1 = \frac{(^1/_2 - \epsilon_0)^2}{3} F_{10}$

$$\alpha_2 = {}^1/_2\, F_{20}$$

$$f_1(\epsilon) = \frac{3}{(^1/_2 - \epsilon_0)^3} (\epsilon - {}^1/_2)^2$$

$$f_2(\epsilon) = \frac{1}{^1/_2 - \epsilon_0}$$

$$g_1(\epsilon) = F_1/F_{10}$$

$$g_2(\epsilon) = F_2/F_{20}$$

$$F_1 = F_1(\delta) = 3\Phi_1(\delta) - \Phi_2(\delta) - 2F(Z)$$

$$F_{10} = F_1(\delta_{min})$$

$$F_2 = F_2(\delta) = {}^3/_2\Phi_1(\delta) + {}^1/_2\Phi_2(\delta) - 2F(Z)$$

$$F_{20} = F_2(\delta_{min})$$

and

$$\delta_{min} = \frac{136m}{Z^1/^3E} \bullet 4$$

is the minimal value of the screening variable (δ). It can be seen easily that the functions $f_i(\epsilon)$ are normalized and that the functions $g_i(\epsilon)$ are "valid" rejection functions.

Therefore, given a set of uniformly distributed random numbers r0,r1,r2,... ($0. \leq ri \leq 1.$) we can sample the variate ϵ (x in the program) by

1) selecting the integer i (1 or 2)

$$BPAR = \frac{\alpha_1}{\alpha_1 + \alpha_2} = \frac{DSIG1}{DSIG1 + DSIG2}$$

if $r_0 <$ BPAR $\quad i=1$

if $r_0 \geq$ BPAR $\quad i=2$

2) sampling ϵ from $f_1(\epsilon)$. This can be performed by the following expressions

$$\epsilon = 0.5[1-(0.5\epsilon_0)(1-r_1)^1/^3] \quad i=1$$

$$\epsilon = \epsilon_0+(^1/_2-\epsilon_0)r_1 \qquad i=2$$

3) calculating the rejection function $g_1(\epsilon)$ and rejecting/accepting ϵ

if $r2>g_i(\epsilon)$ reject ϵ, return to step a.
if $r2\leq g_i(\epsilon)$ accept ϵ.

It should be mentioned that we need a step just after sampling ϵ in the step b. The point is that the cross section formula goes to negative values at sufficiently large δ. The cross section must not be allowed to go negative, so this imposes an upper limit for δ

$$\delta_{max} = \exp[(21.12-F(Z))/4.184] - 0.962$$

If we get a δ value using the sampled ϵ so that $\delta>\delta_{max}$, we have to start again from the step a.

After the successful sampling of ϵ, GPAIR generates the polar angles of the electron/positron with respect to an axis defined along the direction of the parent photon. The azimuthal angle Φ is generated isotropically and Θ is assigned the approximate average value given by Marmier and Sheldon (see Ref. 4) as

$$\Theta = m/E$$

Using this information together with the momentum conservation enables the momentum vectors of both decay products to be calculated and transformed into the GEANT coordinate system.

The choice of which particle in the pair is the electron/positron is made randomly.

2. Restrictions

1) Effects due to the breakdown of Born approximation at low energies are ignored (but the Coulomb correction is included now).

2) As suggested by Ford and Nelson (3), for very low energy photons ($E \leq 2.1$ MeV) the electron energy is approximated by sampling from a uniform distribution over the interval $m \longrightarrow {}^1/_2E$. The reason for this suggestion is that the sampling method used in EGS and in the earlier GEANT versions becomes progressively more inefficient as the pair threshold is approached. This is not true for the sampling method outlined above (the efficiency of the method pratically does not depend on the photon energy), but we have chosen to keep this approximation.

3) Target materials composed of compounds or mixtures are treated identically to elements (this is not the case when computing the mean free path!) using the effective atomic number computed in the routine GSMIXT. It can be show that the error of this type of treatment is small and can be neglected.

4) The differential cross-section implicitly accounts for pair production in both nuclear and atomic electron fields. However, triplet production is not generated, and the recoil momentum of the target nucleus/electron is assumed to be zero.

3. References

1) J.C. Butcher & H. Messel : Nucl. Phys. 20 (1960) 15

2) J.M. Hammersley, D.S. Handscomb : Monte Carlo methods, J. Wiley et Sons Inc. New York, 1964.

3) R. Ford & W. Nelson : SLAC-210, UC-32 (June 1978).

4) P. Marmier & E. Sheldon : Physics of Nuclei and Particles Vol. 1, Academic Press (1969).

GEANT 3.10 | USER'S GUIDE | PHYS 220

Author(s) : G.N.Patrick, L.Urban
Origin : Same

Submitted: 26.10.84
Revised: 30.05.86

Total cross-section for Compton Scattering

CALL GCOMPI

The mean free path is tabuled at initialisation time as a function of the medium and of the energy (see JMATE data structure). The energy binning is set within the array ELOW (COMMON GCMULO) in the routine GPHYSI.

1. Method

The mean free path, λ, for a photon to interact via Compton scattering is given by

$$\lambda = \frac{1}{\Sigma} = \frac{A}{N.\rho.\sigma(Z,E)} \qquad [1]$$

where,

- N Avogadro's number
- Z,A atomic number and weight of the medium
- ρ density of the medium
- σ total cross-section per atom for the compton scattering
- E energy of the photon.

For the total cross-section an empirical cross-section formula is used which produces the cross-section data rather well down to 10 KeV.

$$\sigma(Z,E) = Z \left[p1(Z) \frac{Ln(1+2X)}{X} + \frac{p2(Z)+p3(Z)\bullet X+p4(Z)\bullet X^2}{1+a.X+b.X^2+c.X^3} \right] \text{barn/atom} \qquad [2]$$

where m = electron mass ; X = E/m ; $P_i(Z) = D_i+E_i.Z+F_i.Z^2$

The values of the parameters are put in Data within the routine GCOMPI.

The fit was achieved over 511 data choosen between
$5 \leq Z \leq 100$; $10KeV \leq E \leq 100GeV/Z$

The accuracy of the fit is estimated to:

$\Delta\sigma/\sigma \simeq 10\%$ for $E \simeq 10KeV - 20KeV$

$\Delta\sigma/\sigma \simeq 5-6\%$ for $E > 20KeV$

Author(s) : G.N. Patrick
Origin : Same

Submitted: 26.09.83
Revised: 13.05.86

Simulation of Compton Scattering

CALL GCOMP

GCOMP generates the Compton Scattering of a photon from an atomic electron by using the random number techniques of Butcher and Messell(1) to sample the scattered photon energy from the Klein-Nishina formula(2).
All input/output information is routed through specially constructed GEANT common blocks.

Input:

via COMMON/GCTRAK/....	VECT(1)	x	
	(2)	y	
	(3)	z	
	(4)	P_x/P	> incident
	(5)	P_y/P	photon
	(6)	P_z/P	
	(7)	P	

Output:

via COMMON/GCKING/....	GKIN(1,1)	P_x	
	(2,1)	P_y	
	(3,1)	P_z	> recoil
	(4,1)	P	electron
	(5,1)	particle	
			type (3)
via COMMON/GCTRAK/....	VECT(4)	P_x/P	
	(5)	P_y/P	scattered
	(6)	P_z/P	photon
	(7)	P	

Compton scattering is selected in GEANT by means of the FFREAD data card COMP.

When selected, GCOMP is automatically called from the GEANT gamma tracking routine GTGAMA.

1. Method

A complete account of the Monte-Carlo methods used would be outside the scope of this write-up, and only the basic formalism is outlined. The interested user is recommended to refer to the publications of Butcher and Messel (1), Messel and Crawford (3), and Ford and Nelson (4) for more details.

The quantum-mechanical Klein-Nishina differential cross-section is given by Ford & Nelson (4) as

$$\Phi(E,E') = \frac{X_0 n \pi r_0{}^2 m_e}{E^2} \left[\frac{1}{\varepsilon} + \varepsilon\right] \left[1 - \frac{\varepsilon \sin^2\theta}{1+\varepsilon^2}\right]$$

here,

E = energy of the incident photon
E' = energy of the scattered photon
ε = E'/E
m_e = electron mass
n = electron density
r_0 = electron radius
X_0 = radiation length

Assuming an elastic collision, the scattering angle θ is defined by the Compton formula

$$E' = \frac{Em_e}{m_e + E(1-\cos\theta)}$$

Using the combined "composition" and "rejection" Monte-Carlo methods described in the GPAIRG entry(PHYS 211), we may set:

$$f(\varepsilon) = \left[\frac{1}{\varepsilon} + \varepsilon\right] = \sum_{i=1}^{2} \alpha_i f_i(E) \qquad \text{for } \varepsilon_0 < \varepsilon < 1$$

$$g(\varepsilon) = \left[1 - \frac{\varepsilon \sin^2\theta}{1+\varepsilon^2}\right] \qquad \text{rejection function}$$

$$\alpha_1 = \ln(1/\varepsilon_0)$$

$$\alpha_2 = (1-\varepsilon_0{}^2)/2$$

$$f_1(\varepsilon) = \frac{1}{\ln(1/\varepsilon_0)}(1/\varepsilon)$$

$$f_2(\varepsilon) = \frac{2\varepsilon}{1-\varepsilon_0{}^2}$$

The value of ε corresponding to the minimum possible scattered photon energy (backward scattering) is given by:

$$\varepsilon_0 = \frac{1}{1+2E/m_e}$$

In practice, therefore, given a set of random numbers R0,R1,...,Rn, GCOMP samples the variate ε by:

(a) Deciding which element of the $f(\varepsilon)$ distribution to sample from

α_T = $(\alpha_1+\alpha_2)$.R0

If $\alpha_1 \geq \alpha_T$ select $f_1(\varepsilon)$

$\alpha_2 < \alpha_T$ select $f_2(\varepsilon)$

(b) Sampling ε from the distributions corresponding to f_1 and f_2. For f_1, this is simply achieved by

$$\varepsilon = \varepsilon_0 e^{\alpha_1 . R1}$$

For f_2, we change variables and use

$$\varepsilon' = \mathrm{MAX}(R2,R3) \qquad \text{for } E/m \geq (E/m+1).R4$$
$$\varepsilon' = R5 \qquad \text{for all other cases}$$

Then, $\varepsilon = \varepsilon_0 + (1-\varepsilon_0)\varepsilon'$

(c) Calculating $\sin^2\theta$ and testing the rejection function

$$t = m_e(1-\varepsilon)/E'$$
$$\sin^2\theta = \mathrm{MAX}(0.,t(2-t))$$

If $R6 > g(\varepsilon)$ reject ε
$R6 \leq g(\varepsilon)$ accept ε

After the successful sampling of ε, GCOMP then generates the polar angles of the scattered photon with respect to an axis defined along the parent photon. The azimuthal angle Φ is generated isotropically and Φ is as defined above.

This information is then used to calculate the momentum vector of the scattered photon and recoil electron, and to then transform them into the GEANT coordinate system.

2. Restriction

1) The differential cross-section is strictly only valid for those collisions in which the energy of the recoil electron is large compared with its binding energy (which is ignored). However, as pointed out by Rossi (5), this has a negligible effect because of the small number of recoil electrons produced at very low energies.

3. References

1) J.C. Butcher & H. Messel :	Nucl. Phys. 20 (1960) 15
2) O. Klein & Y. Nishina :	Z. f. Physik 52 (1929) 853.
3) H. Messel & D.F. Crawford :	Electron - Photon Shower Distribution Function Tables for Lead, Copper and Air Absorbers, Pergamon Press (1970) 3.
4) R. Ford & W. Nelson :	SLAC-210 UC-32 (June 1978).
5) B. Rossi	High Energy Particles, Prentice-Hall Inc. (1952) 77−79.

Author(s) : L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 13.05.86

Total cross-section for Photo electric effect

CALL GPHOTI Function GPHSIG

The Mean Free Path is tabulated at initialisation time as a function of the medium and of the energy (see JMATE data structure). The energy binning is set within the array ELOW (COMMON GCMULO) in the routine GPHYSI.

1. Method

The mean free path, λ, for a photon to interact via photoeffect is given by

$$\lambda = 1/\Sigma \quad \text{(in cm)} \qquad [1]$$

where, Σ macroscopic cross-section is given as

$$\Sigma = \frac{N\rho\sigma(Z,E_\gamma)}{A} \quad \text{in (1/cm)} \qquad [2a]$$

$$\Sigma = \frac{N\rho\Sigma_i p_i\sigma(Z_i,E_\gamma)}{\Sigma_i p_i A_i} = N\rho \Sigma_i (w_i/A_i) \bullet \sigma(Z_i,E_\gamma) \qquad [2b]$$

for the chemical element or compound/mixture respectively where,

N	Avogadro's number
$Z(Z_i)$	atomic number (of i-th component) of the medium
$A(A_i)$	atomic number (of i-th component) of the medium
ρ	density
σ	total cross-section
p_i	proportion by number of the i-th element in the material ($p_i \simeq w_i/A_i$ where w_i the corresponding proportion by weight).

For $5 \leq Z \leq 100$ the Binding Energy of the inner shells have been parametrized as:

$$E_i(Z) = Z^2(A_i + B_iZ + C_iZ^2 + D_iZ^3) \text{ MeV} \qquad [3]$$

where, $i = K, L_I, L_{II}$

	A	B	C	D
K	0.66644E−5	0.22077E−6	−0.32552E−8	0.18199E−10
L_I	−0.29179E−6	0.87983E−7	−0.12589E−8	0.69602E−11
L_{II}	−0.68606E−6	0.10078E−6	−0.14496E−8	0.78809E−11

Then the photoeffect cross-section can be parametrized as:

$$\sigma(Z,E) = Z^{\alpha} X^{\beta} \left| \begin{array}{ll} (p_1/Z) + p_2 X + p_3 + p_4 Z + (p_5/X) + p_6 Z^2 + p_7 (Z/X) + (p_8/X^2) & \\ + (p_9 Z^3) + p_{10}(Z^2/X) + p_{11}(Z/X^2) + p_{12}(1/X^3) & \text{if } E > E_k(Z) \\ (p_{13}/Z) + p_{14} X + p_{15} & \text{if } E_{L_I} < E \le E_k(Z) \\ (p_{16}/Z) + p_{17} X + p_{18} & \text{if } E_{L_{II}} < E \le E_{L_I} \\ p_{19} & \text{if } 10\text{KeV} \le E \le E_{L_{II}} \end{array} \right.$$

$$X = m/E \quad ; \quad \sigma \text{ in barn/atom} \qquad [4]$$

The fit was done over 301 data points choosen between

$5 \le Z \le 100$; $10\text{KeV} \le E \le 50\text{MeV}$

The values of the parameters are put in DATA within the function GPHSIG which compute the formulae [4] in barn/atom.

The accuracy of the fit is estimated:

$\Delta\sigma/\sigma \le 25\%$ near to the peaks

$\Delta\sigma/\sigma \le 10\%$ otherwise.

2. References

1) H. Storm, H.I. Israel, Nucl. Data Tables, A7/1970/565.

Author(s) : L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 13.05.86

Simulation of Photo electric effect

CALL GPHOT

The routine GPHOT generates the atomic photoeffect.

Input : via COMMON GCTRAK

Output: via COMMON GCKING.

GPHOT is automatically called by the tracking routines, if, and when, the photon reaches its interaction point.

1. Method

The simulation of the photoeffect is presently rather crude. If photoelectric interaction has occured a photoelectron is created with total energy

$$E = E_\gamma - E_k(Z) + m \qquad [1]$$

where, $E_k(Z)$ is the K, L_I or L_{II} shell energy

and with the same direction as the incident photon.

GEANT 3.10 | USER'S GUIDE | PHYS 240

Author(s) : F. Carminati
Origin : H.C. Fesefeldt

Submitted: 20.12.85
Revised: 26.05.86

The Photon Induced Fission on Heavy Materials

1. Introduction

This set of routines generates the photon induced fission in the nuclei of heavy materials. The code is a straight translation into the GEANT dialect of the corresponding code from the GHEISHA Montecarlo Program. The GHEISHA routines PFISS and CALIM have been used. *

The information contained in this chapter is mainly taken from the Gheisha manual (see note) to which the user is referred.

In fissionable elements the inelastic reaction of photons may be accompanied by the fission of the nucleus. This channel is taken into account only when the nuclear mass is bigger than 230. The fission cross section for Uranium is tabulated and for other elements the parametrization of the A dependence at 3 MeV neutron energy is used:

$$\sigma_f\,(3\text{MeV}) = -67.0 + 38.7 \bullet Z^{4/3}/A$$

which implies that the fission may occur only for $Z \geq 90$.

The fission products are generated in an approximate way. The photon is replaced by a pion of random charge with the same kinetic energy. Then the GUHADR routine is called to generate a pion-nucleus inelastic scattering.

The photo fission is activated by the data card PFIS. If IPFIS = 1 has been specified, secondaries coming from the interaction of the virtual photon with the nucleus will be in the GEANT temporary stack. If IPFIS = 2 has been specified, then the secondary particles will not be generated and the energy of the photon will be summed to DESTEP.

A table with the photo-fission cross sections as a function of the energy is put in a bank at initialization time for all the materials with A bigger than 230. See material bank structure for details.

*) GHEISHA: H.C. Fesefeldt III. Physikalisches Institut der RWTH Aachen Physikzentrum, 5100 Aachen, W. Germany, tel. 0241/807274) version 7.03, released on September, 28th, 1985.
For additional informations see "Simulation of Hadronic Showers, Physics and Applications", PITHA-report 85 − 02, H.C. Fesefeldt, RWTH Aachen.

2. Description of the routines

CALL GPFISI

This routines computes and stores in the appropriate bank the value of the photo-fission cross-sections as a function of the energy for a given material. See JMATE data structure.

CALL GPFIS

This routine is called by GTGAMA every time a photon induced fission has to happen. It generates the final state particles. A call to GUHADR is performed if IPFIS (which is the variable set by the PFIS data card) is equal to 1. If the GHEISHA interface is used, a fission is forced. The secondaries from the photo-fission are always generated if IPFIS is equal to 1, irrespectively from the value of IHADR.

Author(s) : R.Brun, M.Hansroul
Origin : Same

Submitted: 01.10.83
Revised: 28.05.86

Gaussian Multiple Scattering

```
CALL GMUL(VECT,STEP,VOUT*)
```

STEP	Step size
VECT	Initial coordinates, direction cosines, momentum
VOUT	Output coordinates, direction cosines, momentum

The call to this routine is automatically selected by the GEANT tracking routines GTELEC, GTHADR, GTMUON. The algorithm is the following.

- first the particle position and direction are transformed into a local frame of reference in which the new Z axis coincides with the direction of the particle;
- two random deviation angles are computed in two orthogonal projections (i.e. in the new planes XZ and YZ). Each angle is obtained from a gaussian distribution with zero mean value and standard deviation given by the well known formula

ASD = 0.015*(SQRT(L/LR)/Pβ

where
P is the momentum in GEV/C
L is the length of material traversed
LR is the radiation length of the material
β is the particle velocity in units of C

- similarly, the deviations of the particle position are computed in the same two orthogonal projections. However, the correlations between the position and the corresponding angle deviation is taken into account;

let
RAX be the random deviation angle in the XZ projection
RPX be the random number obtained from a gaussian distribution of zero mean and unit standard deviation.

Then the deviation of the position is calculated by the relation

DX = 0.5*(RAX + RPX/SQRT(3))*L*ASD

A similar relation holds for the YZ projection;

– then the modified position and direction of the particle are transformed back from the local frame into the original reference frame.

Gaussian Multiple Scattering Formulae

Let us consider a particle traversing a material of radiation length X_0 and let us compute the angular deviation Θ and the lateral displacement d, projected onto any plane a path length of s.

From Rossi (High Energy Particles 1952), the probability law of Θ and d is given by:

$$F_s(\Theta,d) = A\ \exp\left[-\frac{4}{\Phi_0^2}\left(\frac{\Theta^2}{s} - \frac{3d\Theta}{s^2} + \frac{3d^2}{s^3} \right) \right]$$

where, $\Phi_0^2 = \dfrac{[0.021/p\beta]^2}{X_0}$ (X_0 = radiation length)

p = particle momentum in GeV/c

β = particle velocity in units of c (velocity of ligth)

Marginal laws

in Θ : $f(\Theta) = B \exp\left[-\dfrac{\Theta^2}{2\sigma_\Theta^2} \right]$ with $\sigma_\Theta^2 = 1/2\ \Phi_0^2\ s$

in d : $g(d) = C \exp\left[-\dfrac{d^2}{2\sigma_d^2} \right]$ with $\sigma_d^2 = {}^1/_3\ \sigma_\Theta^2\ s^2 = {}^1/_6\ \Phi_0^2\ s^3$

Defining the reduced variables : $u = y/\sigma_d$ and $v = \Theta/\sigma_\Theta$ the probability law can be written as

$$F_s(u,v) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{u^2 - 2\rho uv + v^2}{2(1-\rho^2)} \right]$$

with $\rho = \sqrt{3/2}$

and the distribution of u, knowing v is given by

$$C_v(u) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left[-\frac{(u-\rho v)^2}{2(1-\rho^2)} \right]$$

which leads to

$$F_s(u,v) = g(v)\ C_v(u)$$

Defining the new variable $u' = u - \rho v$ of variance $\sigma_{u'}^2 = 1-\rho^2$, the probability law is expressed as the product of the marginal laws of the 2 **independent** variables v and u'

$U = \rho v$ is called the regression line of u and v and gives the average of u for a given v.

Application to multiple scattering

$$\rho = \sqrt{3}/2 \;\; ; \; \sigma_{u'} = \sqrt{1-\rho^2} = 0.5$$

let $R_{u'}$
R_v be 2 random numbers distributed as a Gaussian with zero mean and unit standard deviation, then

$$u = \rho\, v + u' = \rho\, R_v + 0.5 R_{u'}$$

and the lateral displacement onto a plane becomes:

$$d = 0.5\,[\, R_v + R_{u'} \,/\, \sqrt{3}\,]\, \sigma_\Theta\, s$$

whereas the angular deviation is:

$$\Theta = R_v\, \sigma_\Theta$$

Author(s) : M.S. Dixit
Origin : Same

Submitted: 17.06.85
Revised: 13.05.86

Molière Scattering

1. Subroutines Called

GMOLI, GMOL, GMOLS, GMOL4.

Molière theory corrected for finite angle scattering ($\sin\tau \neq \tau$) as described by Bethe is used to calculate the effect of multiple Coulomb scattering on the charged particle trajectory.

The angular distribution is given by:

$$\tau f(\tau)\, d\tau = (\sin\tau/\tau)^{1/2}\, \eta f(\eta) d\eta \qquad [1]$$

where, $$f((\eta) = f^{(0)}(\eta) + f^{(1)}(\eta)/B + f^{(2)}(\eta)/B^2 \qquad [2]$$

η is the reduced angle defined by $\eta = \tau/\chi_c\sqrt{B}$

where χ_c, the critical angle of scattering is defined by

$$\chi_c^2 = \frac{4\pi e^4 Z_{inc}{}^2 Z_s N_A \rho t}{W E^2 \beta^4} \qquad [3]$$

here Z_{inc} = incident particle charge,
N_A = Avogadro number
ρ = density
W = molecular weight
E = particle energy (MeV)
and t = total path length in the scatterer **(and not its thickness)**

B is defined by the equation:

$$B - \ln B = \ln\Omega_0$$

where, $$\Omega_0 = \frac{\chi_c{}^2}{e^{2c-1}\chi_\alpha{}^2} \qquad [4]$$

Here c is the Euler's constant and χ_α the atomic electron screening angle given by

$$\chi_\alpha{}^2 = \frac{m_e{}^2 e^4\ 1.13}{p^2\eta^2(0.885)^2}\, e(Z_x - Z_E)/Z_s \qquad [5]$$

Z_s, Z_E and Z_X are defined by

$$Z_s = \sum_{n_i=1}^{N} n_i Z_i (Z_i + \varepsilon)$$

$$Z_E = \sum n_i Z_i (Z_i + \varepsilon) \ln Z_i^{-2/3}$$

$$Z_X = \sum n_i Z_i (Z_i + \varepsilon) \ln\{1 + 3.34 (Z_i Z_{inc}/137.036)^2\} \qquad [6]$$

n_i are the number of atoms of atomic number Z_i in the molecule/mixture.

Also $\varepsilon \simeq 1$ for incident electron/positrons
$= 0$ for other particles.

ε accounts for multiple scattering by atomic electrons.

The component distribution functions are given by

$$\begin{aligned} f^{(0)}(\eta) &= 2e^{-\eta^2} = 2D_0(1,1,-\eta^2) \\ f^{(1)}(\eta) &= 2D_1(2,1,-\eta^2) \\ f^{(2)}(\eta) &= D_2(3,1,-\eta^2) \end{aligned} \qquad [7]$$

where $D_n(a,b,z) = \delta^n/\delta a^n[\Gamma(a)M(a,b,z)]$

M(a,b,z) being Kummer's hypergeometric function.

Integrals of the functions $f^{(1)}$ and $f^{(2)}$ needed for the Monte Carlo can be written down directly in terms of the D functions

$$\int_{\eta}^{\infty} \eta f^{(1)}(\eta) d\eta = D_1(2,1,-\eta^2) - D_1(1,1,-\eta^2) - D_0(1,1,-\eta^2)$$

and

$$\int_{\eta}^{\infty} \eta f^{(2)}(\eta) d\eta = {}^1/_2 D_2(3,1,-\eta^2) - D_2(2,1,-\eta^2) - D_1(2,1,-\eta^2) \qquad [8]$$

We factorize the variables χ_c and Ω_0 into 2 parts. The first part depends on the incident particle energy-momentum and the pathlength in the medium. The second part is a constant of the medium for a given incident particle.

Subroutine GMOLI evaluates the 2 constants for 4 different types of incident particles (electron, other charge 1 particles, charge 1/3 quark and charge 2/3 quark).

Subroutine GMOL calculates the effect of multiple scattering on the direction of the particle using subroutine GMOLS wich does the actual evaluation of the multiple scattering angles. Subrourine GMOL4 is used to invert the integral of Molière distribution function using 4 point continued fraction interpolation.

Molière distribution is for the total scattering angle. A similar expression may be written down for the lateral displacement of the scattered particles. However, the problem of joint angle lateral displacement in the Molière approximation has not been solved. However, for small step size, lateral displacement is of second order and may be neglected.

Restrictions on the Path Length

Restrictions on the step taken for the path length arise from

a) $\Omega_0 \geq 20$ (to stay within the multiple scattering regime)

b) $\chi_c{}^2 B \leq 1$ (width of the Gaussian part less than 1 radian)

c) Maximum permissable lateral displacement.

At present, for light materials (radiation length > 20 cm), the step is calculated so that the maximum displacement of the particle transverse to the track is less than DMAXMS (specified in GSTMED) at 4 standard deviations level. For heavy materials (radiation length < 20 cm), the step size is arbitrarily set to $^1/_2$ the radiation length.

Path Length Correction

At present, no path length correction is made i.e. path length = thickness.

A path length correction may be applied in an approximate manner as described below.

We have from Fermi Eyges theory

$$t = S + {}^1/_2 \int_0^t \overline{\tau^2(t)}\, dt \qquad [9]$$

$\overline{\tau^2(t)}$ is the mean square angle of scattering
S is the straight line step size, and
t is the actual path length.

We have:

$$\overline{\tau^2(t)} = (0.0212\, Z_{inc}/p\beta)^2 \bullet (t/X_0) \qquad [10]$$

$$X_0 = \text{radiation length.}$$

From [9] and [10] we get

$$S = t \left[1 - 1.1236X10^{-4} \frac{Z_{inc}{}^2 t}{p^2\beta^2 X_0}\right] \qquad [11]$$

Equation [11] may be used to make the path length correction.

2. References

1) H.A. Bethe, Phys. Rev. **89**, 1256, 1953.

2) W.T. Scott, Rev. Mod. Phys. **35**, 231, 1963.

3) R.L. Ford and W.R. Nelson, SLAC-210, (1978).

Author(s) : L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 28.05.86

Ionisation Processes Induced by e^-/e^+

1. Principle

Lets call $[d\sigma(E,T)/dT]$ the differential cross-section for the ejection of an electron of kinetic energy T by an incident e^-/e^+ of total energy E moving in a medium of density ρ.

TCUT is the kinematic energy cutoff below which the soft emitted electrons are treated as continuous energy lost by the incident e^+/e^-, and above which they are explicitly generated (DCUTE in the program).

Then the mean value of the energy lost by the incident e^+/e^- due to the soft electrons is:

$$EL(E,TCUT) = \int_0^{TCUT} T \bullet \frac{d\sigma(E,T)}{dT} dT \qquad [1]$$

whereas the total cross-section for the ejection of an electron of energy T > TCUT is:

$$\sigma(E,TCUT) = \int_{TCUT}^{TMAX} \frac{d\sigma(E,T)}{dT} dT \qquad [2]$$

where TMAX is the maximum energy transferable to the free electron:

TMAX = E−m for e^+ ; TMAX = (E−m)/2 for e^- ; m = electron mass

Subroutine GDRELE(EK,CHARGE,JMA,DEDX)

The integration of [1] leads to the Berger-Seltzer formulae (1,2,3):

$$\frac{dE}{dx} = \frac{2\pi r_0{}^2 mn}{\beta^2} \left[\ln \frac{2(\tau+2)}{(I/m)^2} + F\pm(\tau,\Delta) - \delta \right] \qquad [3]$$

where, $\gamma = E/m$; $\beta^2 = 1 - 1/\gamma^2$; $\tau = \gamma - 1$

τ_c = TCUT/m

TCUT energy cut for e± (DCUTE in the program)

τ_{max} maximum possible energy transfer/electron mass

$\tau_{max} = \tau$ for e^+ , $\tau/2$ for e^-

$\Delta = \min(\tau_c, \tau_{max})$

n electron density of the medium
I average mean ionisation energy
δ density effect correction.

The functions F± are given by

$$F^+(\tau,\Delta) = \ln(\tau\Delta) - \frac{\beta^2}{\tau}\left[\tau + 2\Delta - \frac{3\Delta^2 y}{2} - (\Delta - \frac{\Delta^3}{3})y^2 - (\frac{\Delta^2}{2} - \tau\frac{\Delta^3}{3} + \frac{\Delta^4}{4})\quad y^3\right] \qquad [4]$$

$$F^-(\tau,\Delta) = -1 - \beta^2 + \ln[(\tau-\Delta)\Delta] + \tau/(\tau-\Delta) + [\,\Delta^2/2 + (2\tau+1)\ln(1-\Delta/\tau)\,]\ 1/\gamma^2 \qquad [5]$$

where, $y = 1/(\gamma+1)$

The density effect correction is calculated as (4)

$$\delta = \begin{cases} 0 & \text{if } x < x0 \\ 2\ln 10 \bullet X + C + a(X_1 - X)^m & x_0 \leq x \leq x_1 \\ 2\ln 10 \bullet X + C & x_1 < x \end{cases} \qquad [6]$$

where, $x = \ln(\gamma^2 - 1) / 2\ln 10$

The quantities n, I and the parameters of the density effect correction (x_0, x_1, C, a, m) are computed in the routine GPROBI, but we give the corresponding formulae here.

The electron density of the medium, n, can be written as

$$n = \frac{N\rho Z}{A} \qquad \text{for elements}$$

$$n = \frac{N\rho \sum_i p_i Z_i}{\sum_i p_i A_i} \qquad \text{for compounds/mixtures} \qquad [7]$$

where,
N Avogadro's number
$Z(Z_i)$ atomic number (of i-th component) of the medium
$A(A_i)$ atomic weight (of i-th component) of the medium
ρ density of the material
p_i proportion by number of the i-th element in the material (for a mixture p_i can be calculated as $p_i \simeq w_i/A_i$ where w_i the corresponding proportion by weight).

The average mean ionization energy can be calculated as

$$I = (16 \bullet Z^{0.9})10^{-9}\ \text{GeV} \qquad [8]$$

for a chemical element. In the case of a compound or mixture the average value

$$I = \exp[\sum_i p_i Z_i \ln I(Z_i)/\sum p_i Z_i]$$

is used (1,2.3).

The density effect correction parameters can be computed (for condensed medium, 4) as

$$\bar{C} = -C = 1 + 2\ln \frac{I}{28.8\sqrt{\rho(\Sigma p_i Z_i/\Sigma p_i A_i)}10^{-9}}$$

$$m = 3$$

$$X_a = \frac{\bar{C}}{2\ln 10}$$

$$a = \frac{2(\ln 10)(X_a - X_0)}{(X_1 - X_0)^m}$$

I	$\bar{C}$	X_0	X_1
$< 10^{-7}$ $< 10^{-7}$	< 3.681 ≥ 3.681	0.2 $-0.326C - 1$	2 2
$\geq 10^{-7}$ $\geq 10^{-7}$	< 5.215 ≥ 5.215	0.2 $-0.326C - 1.5$	3 3

The subroutine GDRELE(EK,CHARGE,JMA,DEDX) compute the formula [3] above (in GeV/cm) for an electron/positron (CHARGE = −/+1.) of kinetic energy EK in the medium number I:

JMA = LQ(JMATE − I)

Subroutine GDRELA

The energy lost due to the soft delta rays is tabulated at initialisation time as a function of the medium and of the energy by routine GDRELA (see JMATE structure). The energy binning is set within the array ELOW (common GCMULO) in the routine GPHYSI.

The tables are filled with the quantity dE/dX in GeV/cm (formula 3 above) which is valid for an element as well as a mixture or a compound.

In the tables the dE/dX due to ionisation is summed up with the energy lost coming from Bremsstrahlung.

JMA = LQ(JMATE − I)	pointer to the I-th material
JEL1 = LQ(JMA − 1)	pointer for dE/dX for electrons
JEL1 + 90	pointer for dE/dX for positron

Subroutine GDRSGA

The integration of formula [2] gives the total cross-section (3,5):

$$\sigma(Z,E,TCUT) = \frac{2\pi r_0{}^2 mZ}{\beta^2(E-m)} \left[\frac{(\gamma-1)^2}{\gamma^2} \left(\frac{1}{2} - x\right) + \frac{1}{x} - \frac{1}{1-x} - \frac{2\gamma-1}{\gamma^2} \ln\frac{1-x}{x} \right] \qquad [9]$$

for Moller (e^-e^-) and

$$\sigma(Z,E,TCUT) = \frac{2\pi r_0{}^2 mZ}{(E-m)} \left[\frac{1}{\beta^2} \left(\frac{1}{x} - 1\right) + B_1 x + B_2 (1-x) - \frac{B_3}{2} (1-x^2) + \frac{B_4}{3} (1-x^3) \right] \qquad [10]$$

for Bhabha scattering ($e^+ e^-$), where,

$$\gamma = \frac{E}{m}; \qquad \beta^2 = 1 - \frac{1}{\gamma^2}$$

$$x = \frac{TCUT}{E-m}; \qquad y = \frac{1}{\gamma+1}$$

$$B_1 = 2 - y^2 \; ; \qquad B_2 = (1-2y)(3+y^2)$$

$$B_3 = (1-2y)^2 + (1-2y)^3 \; ; \; B_4 = (1-2y)^3$$

The formulae [9] and [10] give the total cross-section of the scattering above the threshold energies

$$T^{thr}_{Moller} = 2TCUT$$

$$T^{thr}_{Bhabha} = TCUT \qquad [11]$$

The macroscopic cross-section Σ is:

$$\Sigma = \frac{N\rho\sigma(Z,E,TCUT)}{A}$$

for a compound or mixture:

$$\Sigma = \frac{N\rho \sum_i p_i \sigma(Z_i,E,TCUT)}{\sum_i p_i A_i} = N\rho \sum_i (w_i/A_i) \bullet \sigma(Z_i,E,TCUT)$$

N Avogadro's number

$Z(Z_i)$ atomic number of the material (i-th component of the material)
$A(A_i)$ atomic weight of the material (i-th component)
ρ density of the material
σ total cross section per atom for discrete delta ray
p_i proportion by number of the i-th element in the material $p_i \simeq w_i/A_i$ where w_i is the corresponding proportion by weight
TCUT energy cut-off (DCUTE in the program).

The mean free path, $\lambda = 1/\Sigma$ (in cm) is tabulated at initialisation time as a function of the medium and of the energy by routine GDRSGA **for electrons and positron.** The energy binning is set within the array ELOW (common GCMULO) in the routine GPHYSI. (See JMATE data structure).

JMA = LQ(JMATE − I)	pointer to the I-th material
JDRAY = LQ(JMA − 11)	pointer to Delta rays cross-sections
JDRAY	pointer for electrons
JDRAY + 90	pointer for positrons
JDRAY + 180	pointer for μ^+ /μ^-

2. References

1) M.J. Berger, S.M. Selzer, NASA-SP-3012 (1964).

2) H.A. Bethe, Ann. Phys. 5 (1930) 325.
F. Bloch, Z. fur Phys. 81 (1933) 363.

3) R.L. Ford, W.R. Nelson, SLAC-210, UC-32 (1978).

4) R.M. Sternheimer, Phys. Rev. B3 (1971) 3681.

5) H. Messel, D.F. Crawford, Electron-Photon Shower Distribution Function, Pergamon press (1970).

Author(s) : L.Urban, D.Ward
Origin : L.Urban

Submitted: 26.10.84
Revised: 26.05.86

Simulation of the Delta ray production

CALL GDRAY

GDRAY generates the discrete Delta ray production.

Input : via COMMON GCTRAK
Output: via COMMON GCKING.

The routine is automatically called from the GEANT tracking routines GTELEC, GTMUON and GTHADR when a charged particle reaches its interaction point.

1. Method

The differential cross section of the δ ray production can be written as (1,2)

$$\frac{d\sigma}{d\epsilon} = \frac{2\pi Z r_0{}^2 m}{\beta^2 (E-m)} \left[\frac{(\gamma-1)^2}{\gamma^2} + \frac{1}{\epsilon} \left(\frac{1}{\epsilon} - \frac{2\gamma-1}{\gamma^2} \right) + \frac{1}{1-\epsilon} \left(\frac{1}{1-\epsilon} - \frac{2\gamma-1}{\gamma^2} \right) \right] \qquad [1a]$$

for the electron/electron (Moller) scattering and

$$\frac{d\sigma}{d\epsilon} = \frac{2\pi Z r_0{}^2 m}{(E-m)} \left[\frac{1}{\beta^2 \epsilon^2} - \frac{B_1}{\epsilon} + B_2 - B_3 \epsilon + B_4 \epsilon^2 \right] \qquad [1b]$$

for the positron-electron (Bhabha) scattering, where

E energy of the incident particle, M its rest mass,

$$\gamma = \frac{E}{M} , \beta^2 = 1 - \frac{1}{\gamma^2}$$

Z atomic number of the medium

$$y = \frac{1}{\gamma + 1} , B_1 = 2 - y^2 , B_2 = (1-2y)(3+y^2)$$

$$B_3 = (1-2y)^2 + (1-2y)^3 , B_4 = (1-2y)^3$$

$$\epsilon = \frac{T}{E-m}$$

where T is the kinematic energy of the scattered electron (of the lower energy in the case of e^-e^- scattering).

The kinematical limits for the variable ϵ are:

$$\text{for } e^-e^- : \epsilon_0 = \frac{TCUT}{E-m} \leq \epsilon \leq \frac{1}{2} \qquad [2a]$$

$$\text{for } e^+ e^- : \epsilon_0 \leq \epsilon \leq 1 \qquad [2b]$$

For the others charged particles the differential cross-section can be written:

$$\frac{d\sigma}{dT} = 2\pi Z r_0^2 m \frac{1}{\beta^2} \frac{1}{T^2} \left[1-\beta^2 \frac{T}{TMAX} \right] \text{ for spin 0 particle} \qquad [3a]$$

$$\frac{d\sigma}{dT} = 2\pi Z r_0^2 m \frac{1}{\beta^2} \frac{1}{T^2} \left[1-\beta^2 \frac{T}{TMAX} + \frac{T^2}{2E^2} \right] \text{ for spin 1/2 particle} \qquad [3b]$$

where TMAX is the maximum energy transferable to the free electron:

$$TMAX = \frac{2m(\gamma^2-1)}{1+2\gamma(m/M)+(m/M)^2} \qquad [4]$$

and TCUT is the energy threshold for the δ-ray emission:

$$TCUT \leq T \leq TMAX$$

Apart from the normalization, the cross section can be written as:

$$\frac{d\sigma}{d\epsilon} = f(\epsilon).g(\epsilon) \qquad [5]$$

where, for e^-e^- scattering:

$$f(\epsilon) = \frac{\epsilon_0}{1-2\epsilon_0} \frac{1}{\epsilon^2} \qquad [6a]$$

$$g(\varepsilon) = \frac{4}{9\gamma^2-10\gamma+5} \quad [(\gamma-1)^2\varepsilon^2-(2\gamma^2+2\gamma-1) \quad \frac{\varepsilon}{1-\varepsilon} + \frac{\gamma^2}{(1-\varepsilon)^2} \quad] \qquad [6b]$$

for the case of the $e^+ e^-$ scattering:

$$f(\epsilon) = \frac{\epsilon_0}{1-\epsilon_0} \frac{1}{\epsilon^2} \qquad [7a]$$

$$g(\epsilon) = \frac{B_0 - B_1\epsilon + B_2\epsilon^2 - B_3\epsilon^3 + B_4\epsilon^4}{B_0 - B_1\epsilon_0 + B_2\epsilon^2{}_0 - B_3\epsilon^3{}_0 + B_4\epsilon^4{}_0} \qquad [7b]$$

Here $B_0 = \gamma^2/(\gamma^2 - 1)$, all the other quantities have been defined above.

For the others charged particles:

$$f(T) = \left(\frac{1}{TCUT} - \frac{1}{TMAX}\right) \bullet \frac{1}{T} \qquad [8a]$$

$$g(T) = 1-\beta^2 \frac{T}{TMAX} + \frac{T^2}{2E^2} \quad \text{(last term for spin } {}^1/_2 \text{ particle only)} \qquad [8b]$$

GDRAY samples the variate ϵ by:

1) sampling ϵ from $f(\epsilon)$

2) calculating the rejection function $g(\epsilon)$ and accepting the sampled ϵ with a probability of $g(\epsilon)$.

After the successful sampling of ϵ, GDRAY generates the polar angles of the scattered electron with respect to an axis defined by the incident particle. The azimuthal angle Φ is generated isotropically, Θ is calculated from the energy-momentum conservation. This information is used to calculate the energy and momentum of both scattered particles and to transform them into the GEANT coordinate system.

2. References

1) H. Messel, D.F. Crawford, Electron-Photon Shower Distribution Function, Pergamon press (1970).

2) R.L. Ford, W.R. Nelson, SLAC-210, UC-32 (1978).

3) Particle data booklet (April 84).

Author(s) : G.N.Patrick
Origin : G.N.Patrick, R.Brun

Submitted: 10.04.86
Revised: 23.05.86

Simulation of Energy Loss Straggling

```
CALL GLANDO(IMODE,STEP,Z,A,RHO,P,E,XMASS,DE,IFLAG)
```

Input :	IMODE	= 2	automatic selection of relevant distribution	
		= 3	Landau (RANLAN) sampling	
		= 4	Landau (GENLAN) sampling	
		= 5	Vavilov (DINVAV) sampling	
		= 6	Gaussian sampling	
	STEP	=	current step−length	(cm)
	Z	=	atomic number of absorber	
	A	=	atomic weight of absorber	
	RHO	=	density of absorber	(g/cm**3)
	P	=	momentum of incident particle	(GeV)
	E	=	energy of incident particle	(GeV)
	XMASS	=	mass of incident particle	(GeV)
Output :	DE	=	DE/DX − <DE/DX>	(GeV)
	IFLAG	= 3	Landau (RANLAN) sampling used	
		= 4	Landau (GENLAN) sampling used	
		= 5	Vavilov (DINVAV) sampling used	
		= 6	Gaussian sampling used	
		= 30	Vavilov arguments out of range	

GLANDO is automatically called by the tracking routines GTELEC, GTHADR and GTMUON when the LOSS data card argument ILOSS is set > 1. The various samplings can be selected by means of this data card since it has a direct correspondence with the IMODE argument above. The default is ILOSS = 2 (automatic selection of distribution).

Warning : For GEANT version 3.10 the Vavilov selection (IMODE = 5) is disabled (see section 2). If IMODE = 2 is selected, the Landau distribution is substituted in the region where the Vavilov distribution would normally be chosen.

1. Method

Due to the statistical nature of ionisation energy loss, large fluctuations can occur in the amount of energy deposited by a given particle traversing a given absorber element. As recently reviewed [1], continuous processes such as multiple scattering and energy loss play a dominant role in the longitudinal and lateral development of electromagnetic & hadronic showers, and in the case of sampling calorimeters the measured resolution can be significantly degraded by such fluctuations in their active layers.

The description of ionisation fluctuations is characterised by the significance parameter, κ, which is proportional to the ratio of mean energy loss to the maximum allowed energy transfer in a single collision with an atomic electron

$$\kappa = \xi/E_{max}$$

E_{max} is the maximum transferable energy in a single collison with an atomic electron.

$$E_{max} = \frac{2m_e^2\eta^2}{1 + 2\gamma m_e/m_x + (m_e/m_x)^2} \qquad \eta = \beta\gamma$$

ξ arises from the Rutherford scattering cross-section and is defined as

$$\xi = \frac{2\pi z^2 e^4 \; N_a Z\rho\delta x}{m_e\beta^2 c^2 \; A}$$

$$= 153.4(z^2/\beta^2)(Z/A)\rho\delta x \;\; \text{keV}$$

For a given absorber, κ can therefore tend towards large values if δx is large and/or if β is small. Likewise, κ can tend towards zero if δx is small or/and if β approaches 1. There are therefore two basic regimes which occur in the description of ionisation fluctuations :

(a) A large number of collisions involving the loss of all or most of the incident particle energy during the traversal of an absorber. As the total energy transfer is composed of a multitude of small energy losses, we can apply the central-limit theorem and describe the fluctuations by a Gaussian distribution. This case is applicable to non-relativistic particles and is described by the inequality

$\kappa > 10$

(ie. when the mean energy loss over the absorber is greater than the maximum energy transfer in a single collision).

(b) Particles traversing thin counters and incident electrons under any conditions. The relevant inequalities and distributions are

$0.01 < \kappa < 10$ ..... Vavilov distribution
$\kappa < 0.01$ ..... Landau distribution

1.1 Landau Theory

For a particle of mass m_x traversing a thickness of material δx the Landau probability distribution may be written in terms of the universal Landau [2] function $\phi(\lambda)$ as

$$f(\varepsilon,\delta x) = 1/\xi \;\phi(\lambda)$$

where,

$$\phi(\lambda) = 1/(2\pi i) \int_{c-i\infty}^{c+i\infty} e^{u \ln u + \lambda u} du$$

$$\lambda = \frac{\varepsilon - <\varepsilon>}{\xi} - \beta^2 - 1 + C - \ln(\xi/E_{max})$$

C = Euler constant from transcendental function theory, i.e.: 0.577215...
$<\varepsilon>$ = average energy loss

ε = actual energy loss

1.2 Vavilov Theory

After Landau's work, Vavilov [3] derived a more accurate straggling distribution by introducing the kinematic limit on the maximum transferable energy in a single collision, rather than using $E_{max} = \infty$.

$$f(\varepsilon,\delta s) = 1/\xi\ \phi_V(\lambda_V,\kappa,\beta^2)$$

where,

$$\phi_V(\lambda_V,\kappa,\beta^2) = \kappa/\pi \exp(\kappa(1+\beta^2 C) \int \exp(\kappa f_1) \cos(u\lambda_V+\kappa f_2)\, du$$

$$\lambda_V = \frac{\varepsilon - <\varepsilon>}{E_{max}} - \kappa(1+\beta^2-C)$$

C = Euler's constant = 0.577216

$$f_1(u) = \beta^2[\log u - Ci(u)] - \cos u - u\, Si(u)$$

$$f_2(u) = u[\log u - Ci(u)] + \sin u + \beta^2 Si(u)$$

$$Si(u) = \int \sin u/u\, du \quad \text{(sine integral)}$$

$$Ci(u) = \int \cos u/u\, du \quad \text{(cosine integral)}$$

$$\lambda_V = \kappa\, [\ 1/\xi(\varepsilon - <\varepsilon>)-1-\beta^2+C]$$

The Vavilov parameters are simply related to the Landau parameter λ by

$$\lambda = \lambda_V/\kappa - \ln\kappa$$

It can be shown that as $\kappa \rightarrow 0$ the Vavilov distribution approaches that of Landau and for $\kappa \rightarrow \infty$ the Vavilov distribution tends to a Gaussian distribution [4].

1.3 Gaussian Theory

Various conflicting forms have been proposed for Gaussian straggling functions, but most of these appear to have little theoretical or experimental basis. However, as noted by Schorr [5] it has been demonstrated by Seltzer and Berger [6] that for $\kappa \geq 10.0$ the Vavilov distribution can be replaced by a Gaussian of the form :

$$f(\varepsilon,\delta s) \simeq \frac{1}{\xi \sqrt{(2\pi)/\kappa}\ (1-\beta^2/2)} \exp\{-(\varepsilon - <\varepsilon>)^2\ \kappa/2\xi^2(1-\beta^2/2)\}$$

thus implying .... mean = $<\varepsilon>$

$$\sigma^2 = \xi(1-\beta^2/2)\ E_{max}$$

2. Implementation

We have utilised two routines from the CERN Program Library to sample random numbers X from the Landau distribution :

(a) RANLAN written by Kolbig [7].(copied into GEANT Pam File as GLANDR)

```
    X = RANLAN(RNDM(0))
```

(b) GENLAN written by James and Hancock [8].(copied into GEANT Pam File as GLANG)

```
    CALL GENLAN(X)
```

It should be noted that over the years considerable confusion has arisen over the precise form and features of the Landau distribution. As an example, $\phi(\lambda)$ has been quoted as having it's maximum at $\lambda = 0.225$ {Fano [9]} and $\lambda = -0.05$ {Landau [2]}, whereas the true value is in fact $\lambda = -0.222782...$ {Kolbig & Schorr [10]}.

For the Vavilov distribution we have used the function DISVAV written by Schorr [11]. This is called as follows :

```
CALL COEDIN(RKA,BE2,J)
      .
      .
      .
X     = DINVAV(RNDM(0))
```

where, RKA = κ, BE2 = β^2 and J= print flag.

In this case there is a disadvantage for Monte-Carlo tracking programs in that every time κ or β changes, COEDIN must in principle be called again. Since, even in a constant medium, β and δx will vary from one tracking step to another, frequent COEDIN calls will be incurred with heavy time penalties. For this reason we have currently disabled the Vavilov option and optimisation studies are underway [12].

3. Restrictions

The Landau formalism makes two restrictive assumptions :

(a) The typical energy loss in the absorber should be large compared to the binding energy of the most tightly bound electron. At energies > 1 MeV this assumption is satisfied and at low energies this restriction is removed in the Vavilov theory.

(b) The typical energy loss is small compared to the maximum energy loss in a single collision. Unfortunately, for gaseous detectors, typical energy losses are few keV which is comparable to the binding energies of the inner electrons. In such cases a more sophisticated approach which accounts for atomic energy levels (see for instance Talman[13]) is necessary to accurately simulate data distributions.

As noted by Nelson et al [14], one should ideally simulate a **restricted** straggling distribution in much the same way that the restricted Bethe-Bloch formula is used for mean energy loss. This is because high energy transfers are often simulated discretely (and therefore separately) in the form of δ-ray production, and not as continuous energy loss. In order to do this analytically one would have to integrate the straggling functions upto the chosen energy cut-off for δ-ray production. Currently, we have adopted a simplistic approach whereby, to avoid double counting, only discrete δ-ray simulation or continuous straggling is allowed, but not both. For this reason, if the LOSS data card is set to be > 1 the option IDRAY becomes disabled (see BASE 040).

4. References

1. G.N. Patrick :	Physical Processes in the Simulation of Electromagetic Cascades, Proc. Workshop on Shower Simulation for LEP Experiments, CERN, 29-31 January 1985
2. L. Landau :	On the Energy Loss of Fast Particles by Ionisation, J. Phys. (USSR) 8(1944), 201
3. P.V. Vavilov :	Ionisation Losses of High-Energy Heavy Particles, Soviet Physics JETP 5(1957), 749
4. B. Schorr :	The Vavilov Distribution and the Landau Parameter λ, private communication
5. B. Schorr :	Programs for the Landau and the Vavilov Distributions and the Corresponding Random Numbers, Comp. Phys. Comm. 7(1974), 216
6. S.M. Seltzer & : M.J. Berger	Energy Loss Straggling of Protons and Mesons : Tabulation of the Vavilov Distribution, Studies in Penetration of Charged Particles in Matter, Nuclear Science Series 39(1964), Nat. Academy of Sciences, Washington D.C.
7. K.S. Kolbig :	Landau Distribution, CERN Program Library, G110, Short Write-up
8. F. James & : S. Hancock	Generation of Landau-Distributed Random Numbers, CERN Program Library,GENLAN, G903, Short Write-up
9. U. Fano :	Penetration of Protons, Alpha Particles and Mesons, Ann. Rev. Nucl. Sci., 13(1963), 1
10. K.S. Kolbig & : B. Schorr	A Program Package for the Landau Distribution, Comp. Phys. Comm. 31(1984), 97
11. B. Schorr :	Programs for the Vavilov Distribution and the Corresponding Random Numbers, CERN Program Library, G111, Long Write-up.
12. G.N. Patrick :	Energy Loss Straggling using Landau, Vavilov, and Gaussian Distributions, in preparation
13. R. Talman :	On the Statistics of Particle Identification using Particle Identification using Ionization, Nucl. Inst. & Meth. 159(1979), 189.
14. W.R. Nelson et al :	The EGS4 Code System, SLAC-265 (1985)

Author(s) : M. Maire
Origin : Same

Submitted: 30.05.86
Revised:

Energy Threshold for Delta Ray production

```
CALL GDRCUT(ITMED,IELOS,XDRAY,DRCUT)
```

Landau fluctuations or delta ray production ? Which values for DCUTE and DCUTM ?

Input : ITMED tracking medium number
XDRAY number of produced delta rays by cm desired by the user.

Output: 1) IELOS = 1 delta ray production suggested
then DRCUT = energy threshold for delta ray production (in GeV)

2) IELOS = 2 Landau fluctuation suggested.
then DRCUT = BIG

The user should be aware that this routine has no action on the GEANT system. It is just an utility in order to help the user to take a decision, for instance via the routine GSTPAR [CONS 210].

1. Method

When a charged particle is moving in a given material, there are two ways to simulate the energy loss by ionisation.

> The first way consists in calculating the mean value of the energy loss (DE/DX) with the full Bethe–Bloch equation and to simulate the fluctuations around this mean value from a Landau distribution [PHYS 332]. In GEANT this method is selected by the flag ILOSS ≥ 2
>
> The second way consists in taking account the fluctuations by generating explicitly the delta rays **above** a given threshold (DCUTE for electrons, DCUTM for others particles) [PHYS330, 430]. In this case the mean value of energy loss must be computed from a restricted energy loss formula (1), and both DE/DX and the number of delta rays produced are function of this cut: DRCUT.

Prescriptions: Concerning the distribution of the energy loss by the incident particle, the two methods are equivalent if the total number of delta rays generated along the trajectory is statistically sufficient: i.e. a few ten along the full trajectory of the particle in the given media.

For a relativistic particle, the number of delta rays produced by cm can be estimated by:

$$\frac{dN}{dX} \simeq \frac{D}{2} \bullet \frac{\rho Z}{A} \bullet \frac{1}{DRCUT} \qquad [1]$$

where D = 0.307 MeV.cm^2/g
ρ = density of the medium
Z,A = atomic number and atomic weight of the medium
DRCUT = energy threshold for emitted delta rays.

This formula holds for electrons/positrons as well as for others particles.

The number of delta rays produced must be sufficient in order to give a Landau distribution of the energy loss by the incident particle, but not too large as the timing for tracking such delta rays will increase dramatically without any improvement of the energy loss distribution.

It is of the responsability of the user to estimate the number of delta rays produced by cm that he would like, according the prescription underlined above. Then the routine GDRCUT returns the corresponding energy threshold DRCUT computed from formula (1). It is recommended to use this value for DCUTE and DCUTM in the current medium.

However, the GEANT material tables are not garantee below ELOW = 10KeV. Therefore, DRCUT can not be smaller than ELOW. Usually, in a non dense material like gas, the number of emitted delta rays for DRCUT = 10KeV will not be sufficient to ensure a correct energy loss distribution. In that case the explicit delta ray generation is not recommended and the Landau fluctuations must be used instead [PHYS 332]. The routine GDRCUT takes care of such cases.

2. Default values in GEANT3

(See BASE 040 Section 4.)

In order to avoid double counting, an automatic protection is put within the code: **if ILOSS = 2 one cannot change the default values of delta rays generation.**

The Table below summarize the different cases.

	Default parameters for delta rays generation	
	Landau ON LOSS = 2 (default)	Landau OFF LOSS = 1
IDRAY DCUTE DCUTM	0 10^4GeV 10^4GeV	2 CUTELE CUTELE

3. References

1) Review of particles properties, April 1984.

2) See also GEANT3 routines : GDRELA, GDRSGA, GLANDO, GSTPAR.

Author(s) : L.Urban
Origin : Same

Submitted: 28.05.86
Revised: 22.07.87

Total cross-section and energy loss for Bremsstrahlung by e^-/e^+

1. Method

Lets call [$d\sigma(Z,T,k)/dk$] the differential cross-section for production of a photon of energy k by an electron of kinetic T in the field of an atom of charge Z

k_c is the energy cutoff below which the soft photons are treated as continuous energy lost **(BCUTE in the program).**

Then the mean value of the energy lost by the electron due the to soft photons is

$$EL(Z,T,k_c) = \int_0^{k_c} k \, \frac{d\sigma(Z,T,k)}{dk} \, dk \qquad [1]$$

whereas the total cross-section for the emission of a photon of energy $> k_c$

$$\sigma(Z,T,k_c) = \int_{k_c}^{T} \frac{d\sigma(Z,T,k)}{dk} \, dk \qquad [2]$$

Many theories of the Bremsstrahlung process exist, each with its own limitations and regions of applicability. Perhaps the best syntesis of these theories can be found in the paper of S.M. Seltzer and M.J. Berger [Ref. 1]. The authors give a tabulation of the Bremsstrahlung cross-section $d\sigma/dk$, differential in the photon energy k, for electrons with kinetic energies T from 1 KeV to 10 GeV. For electron energies above 10 GeV the screened Bethe-Heitler differential cross-section can be used [Ref. 2] together with the Midgal corrections [Refs. 3 and 4]. The first of the two Migdal corrections is important for very high electron energies only ($T \geq 1$ TeV) and has the effect of reducing the cross-section. The second Migdal correction is effective even at "ordinary" energies (100 MeV $\simeq$ 1 GeV) and it decreases the differential cross-section at photon energies below a certain fraction of the incident electron energy ($d\sigma/dk$ decreases significantly if $k/T \leq 10^{-4}$).

Using the tabulated cross-section values of Seltzer and Berger together with the Migdal corrected Bethe-Heitler formula we have computed $\sigma(Z,T,k_c)$ and we have used these computed values as "data points" in the fitting procedure Calculating the "low energy" ($T \leq 10$ GeV) data we have applied the second Midgal correction to the results of Seltzer and Berger.

We have choosen the parametrisations:

$$\sigma(Z,T,k_c) = \frac{Z(Z+\xi_\sigma)(T+m)^2}{T(T+2m)} \, [\, ln(T/k_c)\,]^{\alpha} \bullet F_\sigma(Z,X,Y) \quad (barn) \qquad [3]$$

and

$$EL(Z,T,k_c) = \frac{Z(Z+\xi_l)(T+m)^2}{(T+2m)} \left[\frac{k_c C_M}{T} \right]^{\beta} F_{EL}(Z,X,Y) \quad (GeV\ barn) \qquad [4]$$

where m is the mass of the electron,

$X = \ln(E/m)$, $Y = \ln\,(v_\sigma E/k_c)$ for σ

$X = \ln(T/m)$, $Y = \ln(k_c/v_1 E)$ for EL

with $E=T+m$

The constants ξ_σ, ξ_1, α, β, v_σ, v_1 are parameters to be fitted.

$$C_M = \frac{1}{1+\dfrac{n\, r_0 \bar{\lambda}^2\,(T+m)^2}{\pi k_c^2}}$$

is the Midgal correction factor, with

r_0 classical electron radius reduced compton wavelength of the electron

ƛ = reduced compton wavelength of the electron

n electron density in the medium.

(Seltzer and Berger computed the scaled cross-section

$$\begin{aligned} f(k/T) &= (\beta^2/Z^2)k\,(d\sigma/dk) \\ &= \frac{T(T+2m)}{(T+m)^2\,Z^2}\,k\,(d\sigma/dk)\ , \end{aligned}$$

the factors $(T+m)^2/T(T+2m)$ and $(T+m)^2/T+2m$ come from here).

The functions $F_i(Z,X,Y)$ ($i=\sigma$,EL) have the form

$$F_i(Z,X,Y) = F_{i_0}(X,Y) + ZF_{i_1}(X,Y) \qquad [5]$$

where $F_{ij}(X,Y)$ are polynomials of their variables X, Y:

$$\begin{aligned} Fi_0(X,Y) &= (C_1+C_2X+....+C_6X^5)+(C_7+C_8X+....+C_{12}X^5)y \\ &\quad +(C_{13}+C_{14}X+....+C_{18}X^5)Y^2+....+(C_{31}+C_{32}X+....+C_{36}X^5)Y^5 \qquad y\leq 0 \qquad [6] \\ &= (C_1+C_2X+....+C_6X^5)+(C_7+C_8X+....+C_{12}X^5)Y \\ &\quad +(C_{37}+C_{38}X+....+C_{42}X^5)Y^2+....+(C_{55}+C_{56}X+....+C_{60}X^5)Y^5 \qquad Y>0 \end{aligned}$$

$$
\begin{aligned}
Fi_1(X,Y) \quad &= (C_{61}+C_{62}X+....+C_{65}X^4)+(C_{66}+C{\downarrow}6_7X+....+C_{70}X^4)y \\
&\quad +(C_{71}+C_{72}X+....+C_{75}X^4)Y^2+....+(C_{81}+C_{32}X+....+C_{85}X^4)Y^4 \qquad y\leq 0 \\
&= (C_{61}+C_{62}X+....+C_{65}X^4)+(C_{66}\ +C_{67}X+....+C_{70}X^4)Y \\
&\quad +(C_{86}+C_{87}X+....+C_{90}X^4)Y^2+....+(C_{96}+C_{97}X+....+C_{100}X^4)Y^4 \qquad Y>0
\end{aligned}
\qquad [7]
$$

$F_{ij}(X,Y)$ denotes in fact a function constructed from two polynomials

$$
\begin{aligned}
&P_{ij}^{neg}(X,Y) \quad \text{for } Y\leq 0 \\
&P_{ij}^{pos}(X,Y) \quad \text{for } Y>0
\end{aligned}
\qquad [8]
$$

where the polynomials P_{ij} fulfill the conditions

$$P_{ij}^{neg}(X,Y=0) \ = \ P_{ij}^{pos}(X,Y=0) \qquad [9]$$

$$(\delta P_{ij}^{neg}/\delta Y)\ Y=0 \ = \ (\delta P_{ij}^{pos}\delta Y)\ Y=0 \qquad [10]$$

We have computed 4000 "data points" in the range

Z = 6 ; 13 ; 29 ; 47 ; 74 ; 92

10KeV ≤ T ≤ 10TeV

10KeV ≤ k_c ≤ T

and we have performed a least squares fit to determine the parameters.

The values of the parameters ($\xi\sigma$, α, $v\sigma$, C_i for σ and ξ_1, β, V_1, C for EL) can be found in DATA within the functions GBRSGE and GBRELE which compute the formula [3] and [4] respectively.

The errors of the parametrisations (3) and (4) can be estimated as

$\frac{\Delta\sigma}{\sigma}$	12−15%	for T ≤ 1 MeV
	≤ 5−6%	for 1 MeV < T ≤ 10TeV
$\frac{\Delta(EL)}{EL}$	10−15%	for T≤1 MeV
	5−6%	for 1MeV < T ≤ 100 GeV
	10%	for 100 GeV < T ≤ 10 TeV

We have performed a fit for the "data" without the Midgal corrections too. In this case we used the data of Seltzer and Berger without any correction for T ≤10GeV and we used the Bethe-Heitler cross-section for T ≥ 10GeV. The parametrised forms of the cross-section and energy loss are the same as it was in the first fit (i.e. (3) and (4)), only the numerical values of the parameters have changed. These values are in DATA statements in the functions GBRSGE and GBRELE and this second kind of parametrisation can be activated using the PATCHY switch +USE,BETHE. (The two parametrisations give different results for high electron energy).

CALL GBRELA

The energy loss due to soft photon Bremsstrahlung is tabulated at initialisation time as a function of the medium and of the energy by routine GBRELA (see JMATE data structure). The energy binning is set within the array ELOW (COMMON CGMULO) in the routine GPHYSI.

$$\frac{dE}{dX} = \frac{N\rho}{A} \bullet EL(Z,T,k_c) \quad \text{in GeV/cm} \qquad [11]$$

N	Avogadro's number
A	atomic weight of the material
ρ	density of the material
EL	formulae [4] above.

For a molecule or a mixture:

$$\frac{dE}{dX} = \quad \rho \bullet \Sigma \; w_i \; \frac{1}{\rho_i} \bullet (dE/dX)_i \qquad [12]$$

w_i proportion by weight of the i-th element.

In the Tables, the dE/dX due to Bremsstrahlung is summed up with the energy lost coming from ionisation.

JMA = LQ(JMATE−I)	pointer to the I-th material
JEL1 = LQ(JMA−1)	pointer to dE/dX for electrons
JEL1+90	pointer to dE/dX for positrons.

CALL GBRSGA

The mean free path, λ, for an electron to radiate a photon via Bremsstrahlung is given by

$$\lambda = 1/\Sigma \qquad [13]$$

where Σ is the macroscopic cross-section (1/cm). This macroscopic cross section can be written as

$$\Sigma = \qquad \frac{N\rho\sigma(Z,E,k_c)}{A} \qquad [14a]$$

for a compound or mixture:

$$\Sigma = \qquad \frac{N\rho\Sigma_i p_i\sigma(Z_i,E,k_c)}{\Sigma_i p_i A_i} \quad = \qquad N\rho \sum_i (w_i/A_i) \bullet \sigma(Z_i,E,k_c) \qquad [14b]$$

N	Avogadro's number
$Z(Z_i)$	atomic number of the material (i-th component of the material)

$A(A_i)$ atomic weight of the material (i-th component)

ρ density of the material

σ . total cross section per atom for discrete (photon energy = k_c) Bremsstrahlung (formula [3] above)

p_i proportion by number of the i-th element in the material $p_i \simeq w_i/A_i$ where w_i is the corresponding proportion by weight

k_c photon cut-off energy (BCUTE in the program).

This mean free path is tabuled at initialisation time as a function of the medium and of the energy by routine GBRSGA (see JMATE data structure). The energy binning is set within the array ELOW (COMMON GCMULO) in the routine GPHYSI.

JMA = LQ(JMATE−I)	pointer to the I-th material
JBREM = LQ(JMA−9)	pointer to Brems cross-sections
JBREM	pointer for e^+/e^-
JBREM+90	pointer for μ^+/μ^-

2. References

1) S.M. Seltzer, M.J. Berger : NIM, B12, 95 (1985).

2) R.L. Ford, W.R. Nelson : SLAC-210 4C-32 (1978).
W.R. Nelson, H. Hirayawa, D.W.O. Rogers : SLAC-265 UC-32 (1985).

3) A.B. Midgal : Phys. Rev. **103**, 1811 (1956).

4) H. Messel, D.F. Crawford : Electroni-Photon Shower Distribution Functions (Pergamon Press, Oxford, 1970).

Author(s) : G.N. Patrick, L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 22.07.87

Simulation of discrete Bremsstrahlung by electrons

CALL GBREME

GBREME generates photon Bremsstrahlung by electron by treating it as a discrete process.

The secondary photon energy is sampled from a parametrisation of the Bremsstrahlung calculation of Seltzer and Berger [Ref. 5] for electron energies below 10 GeV, and from the screened Bethe-Heitler cross-section above 10 GeV.

Midgal corrections are applied in both cases.

Input : via COMMON GCTRAK
Output: via COMMON GCKING.

GBREME itself is automatically called from the tracking routines if, and when the parent electron reaches its radiation point during the tracking stage of GEANT.

1. Method

The Coulomb corrected Bethe-Heitler differential cross-section for production of a photon of energy εE by an electron of primary energy E is given as [3]:

$$\frac{d\sigma(Z,E,\varepsilon)}{d\varepsilon} = \frac{r_o^2\alpha Z[Z+\xi(Z)]}{\varepsilon}\{[1+(1-\varepsilon)^2][\Phi_1(\delta)-F(Z)]-\frac{2}{3}(1-\varepsilon)[\Phi_2(\delta)-F(Z)]\} \qquad [1]$$

where, $\Phi_i(\delta)$ are the screening functions depending on the screening variable δ

$$\delta = \frac{136m}{Z^{1/3}E}\ \frac{\varepsilon}{(1-\varepsilon)} \qquad m = \text{electron mass}$$

$$\left.\begin{array}{l}\Phi_1(\delta) = 20.867 - 3.242\delta + 0.625\delta^2 \\ \Phi_2(\delta) = 20.209 - 1.930\delta - 0.086\delta^2\end{array}\right\} \delta \leq 1$$

$$\Phi_1(\delta) = \Phi_2(\delta) = 21.12 - 4.184\ln(\delta+0.952) \qquad \delta > 1$$

$$F(Z) = \begin{cases} {}^4/_3\ln Z & E<0.05\text{GeV} \\ {}^4/_3\ln Z + 4f_c(Z) & E\geq 0.05\text{GeV}\end{cases}$$

$$\varepsilon(Z) = \frac{\ln \dfrac{1440}{Z^{2/3}}}{\ln \dfrac{183}{Z^{1/3}} - f_c(Z)}$$

$f_c(Z)$ the Coulomb correction function

$$\begin{aligned} f_c(Z) &= a*(1/(1+a)+0.20206-0.0369*a+0.0083*a^2-0.002*a^3) \\ a &= (\alpha*Z)^2 \\ \alpha &= 1./137. \end{aligned}$$

The kinematically allowed region for E is

$$\epsilon_c = \frac{k_c}{E} \leq \epsilon \leq 1 - \frac{m}{E} = \epsilon 1 \qquad [2]$$

where, k_c is the photon cut off energy below which the Bremsstrahlun treated as a continuous energy loss (BCUTE in the program).

The cross-section [1] can be decomposed as (the normalisation is not important !)

$$\frac{d\sigma}{d\varepsilon} = \sum_{i=1}^{2} \alpha_i f_i(\varepsilon) g_i(\varepsilon) \qquad [3]$$

where

$$\alpha_1 = F_{10} \ln \frac{\varepsilon l}{\varepsilon c} \qquad \alpha_2 = F_{20} \frac{3}{8} \frac{\varepsilon_1^2 - \varepsilon_c^2}{1 - \varepsilon c}$$

$$f_1(\varepsilon) = \frac{1}{\ln \dfrac{\varepsilon l}{\varepsilon c}} \frac{1}{\varepsilon} \qquad f_2(\varepsilon) = 2 \frac{\varepsilon}{\varepsilon l^2 - \varepsilon c^2}$$

$$g_1(\varepsilon) = \frac{F_1}{F_{10}} \frac{1-\varepsilon}{1-\varepsilon_c} \bullet C_M(\varepsilon) \qquad g_2(\varepsilon) = \frac{F_2}{F_{20}} \bullet C_M(\varepsilon)$$

and

$$F_1(\delta) = 3\Phi_1(\delta) - \Phi_2(\delta) - 2F(Z) \qquad F_{10} = F_1(\delta_{\text{min}})$$

$$F_2(\delta) = 2\Phi_1(\delta) - 2F(Z) \qquad F_{20} = F_2(\delta_{\text{min}})$$

$$\delta_{\text{min}} = \frac{136m}{Z^1/^3E} \frac{\epsilon c}{1 - \epsilon c}$$

is the minimum value of the variable δ.

$C_M(\epsilon)$ is the Midgal correction factor (6):

$$C_M(\varepsilon) = \frac{1 + C_0/\varepsilon_1^2}{1 + C_0/\varepsilon^2}$$

where

$$C_0 = \frac{n r_0 ƛ^2}{\pi}$$

n electron density in the medium
r_0 classical electron radius
$ƛ$ reduced Compton wavelength of the electron.

This correction decreases the cross-section for low photon energy.

Using the decomposition [3] the sampling of ϵ can be done by (r_0, r_1, r_2, etc. a set of random numbers; $0 \leq r_i \leq 1$)

1) selecting the integer i (1 or 2)

   if $r_0 \leq \mathrm{BPAR} = \alpha_1/(\alpha_1+\alpha_2)$ $\quad i=1$

   if $r_0 > \mathrm{BPAR}$ $\quad i=2$

2) sampling ϵ from $f_i(\epsilon)$

   $\epsilon = \epsilon c(\epsilon 1/\epsilon c)^{r_1}$ $\quad i=1$
   $\varepsilon = \sqrt{\varepsilon c^2 + r_1(\varepsilon m^2 - \varepsilon c^2)}$ $\quad i=2$

3) calculating the rejection function $g_i(\epsilon)$ and

   if $r_2 > g_{i(\epsilon)}$ rejecting ϵ, starting again from 1

   if $r_2 \leq g_i(\epsilon)$ accepting ϵ.

As in the case of the pair production after sampling ϵ from $f_i(\epsilon)$ we have to check that

$$\delta = \frac{136m}{Z^{1/3}E}\ \frac{\varepsilon}{1-\varepsilon c} \leq \delta_{\max} = \exp\left(\frac{21.12-f(Z)}{4.184}\right) - 0.952$$

and if this is not the case, we have to start again from step 1.

The decomposition [3] used here is more simple and more effective than that is used in the earlier versions of GEANT and in EGS, and the method outlined above has no convergence problems.

After the successful sampling of ϵ, GBREME generates the polar angles of the radiated photon with respect to an axis defined along the parent electron's momentum. The azimuthal angle Φ is generated isotropically and Θ is assigned the approximate average value given by Marmier and Sheldon [4]

$$\Theta = m/E$$

This information is used to calculate the momentum vector of the radiated photon, transform it to the GEANT coordinate system and store the result into /GCKING/. Also, the momentum of the parent electron is corrected.

2. Restrictions

1) Effects due to breakdown of the born approximation at low energies are ignored.

2) Target materials composed of compounds or mixtures are treated identically to elements using the effective atomic number Z calculated in GSMIXT (this is not the case when computing the mean free path!).

3. References

1) R.W. Williams : Fundamental Formulas of Physics Vol. 2, ed. D.H. Menzel, Dover Pubs. Inc (1960) 550.

2) J.C. Butcher & H. Messel : Nucl. Phys. 20 (1960) 15

3) R. Ford & W. Nelson : SLAC-210, UC-32 (June 1978).

4) P. Marmier & E. Sheldon : Physics of Nuclei and Particles Vol. 1, Academic Press (1969) 140.

5) S. Seltzer and M. Berger: NIM, **B12**, 95 (1985).

6) A.B. Midgal, Phys. Rev. 103, 1811 (1956).

Author(s) : L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 25.04.86

Total cross-section for e^+ e^- Annihilation

CALL GANNII

The mean free path is tabuled at initialisation time as a function of the medium and of the energy (see JMATE data structure). The energy binning is set within the array ELOW (COMMON GCMULO) in the routine GPHYSI.

1. Method

The mean free path, λ, for a positron to interact via annihilation is

$$\lambda = \frac{1}{\Sigma} = \frac{A}{N.\rho.\sigma(Z,E)} \qquad [1]$$

where, Σ macroscopic cross-section (in 1./cm)
N Avogadro's number
Z,A atomic number and weight of the medium
ρ density of the medium
σ(Z,E) total annihilation cross-section per atom.

For the annihilation cross-section the formula of Heitler (see Refs 1 and 2) is used.

The total cross-section are given by ([1],[2])

$$\sigma(Z,E,Ec) = \frac{Z\pi r_0{}^2}{\gamma+1} \left[\frac{(\gamma^2+4\gamma+1}{\gamma^2-1} \ln(\gamma+\sqrt{\gamma^2-1}) - \frac{\gamma+3}{\sqrt{\gamma^2-1}} \right] \qquad [2]$$

where r_0 classical radius of the electron and $\gamma = E/m$

In case of compound/mixture the effective atomic number and weight (evaluated in GSMIXT) is used computing the cross-section. This treatment of the compounds/mixtures is correct in the processes where the cross-section depend on the atomic number Z linearly (annihilation, Moller and Bhabha scattering, Compton scattering).

2. References

1) W. Heitler, The Quantum Theory of radiation, Clarendon Press, Oxford (1954).

2) R.L. Ford, W.R. Nelson, SLAC-210, UC-32 (1978).

Author(s) : L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 29.05.86

Simulation of $e^+ \; e^-$ Annihilation

CALL GANNI

GANNI generates the two-photon annihilation of a positron.

Input : via COMMON GCTRAK
Output: via COMMON GCKING.

The routine is automatically called, if the positron reches its interaction point during the tracking.

1. Method

The differential cross section of the two-photon positron-electron annihilation can be written as [1],[2]:

$$\frac{d\sigma(Z,\epsilon)}{d\epsilon} = ma\,[S(a\epsilon)+S(a(1-\epsilon))] \qquad [1]$$

where, m electron mass

$\gamma = E/m$; E positron energy

$a = \gamma+1$

$\epsilon = k/(E+m)$; k energy of the secondary photon of the lower energy

$$S(x) = C_1\,[\,-1+\frac{C_2}{x}-\frac{1}{x^2}\,]$$

$$C_1 = \frac{Z\pi r_0^{\,2}}{a(E-m)} \quad ; \quad c_2 = a+\frac{e\gamma}{a}$$

Z atomic number of the material.

The kinematical limits for the variable ϵ are given by

$$\varepsilon_0 = \frac{1}{a+\sqrt{\gamma^2-1}} \le \varepsilon \le {}^1/_2 \qquad [2]$$

Due to the symmetry of the formula [1] in ϵ, the range of ϵ can be expanded from $(\epsilon_0, {}^1/_2)$ to

$(\epsilon_0, 1-\epsilon_0)$ and the second function S can be thrown away from the formula. Having done this the differential cross section can be decomposed (apart from the normalisation) as

$$\frac{d\sigma}{d\epsilon} = \underbrace{\frac{1}{\ln\frac{1-\epsilon_0}{\epsilon_0}}\frac{1}{\epsilon}}_{f(\epsilon)} \bullet \underbrace{\frac{(a^2+2a-2)-a^2\epsilon-{}^1/^2}{a^2-2}}_{g(\epsilon)} \qquad [3]$$

Using the expression [3] the secondary photon energy can be sampled by (r_0,r_1 random numbers).

1) sampling ϵ from $f(\epsilon)$

$$\epsilon = \epsilon_0 \exp \left[\ln(\frac{1-\epsilon_0}{\epsilon_0}) \bullet r_0 \right]$$

2) computing the rejection function $g(\epsilon)$ and

   if $r_1 \leq g(\epsilon)$ accepting ϵ
   if $r_1 > g(\epsilon)$ accepting ϵ, starting again from 1.

After the successful sampling of ϵ, the photon energy is computed as

$$k = (E+m)\epsilon \qquad [4]$$

and then GANNI generates the polar angles of the photon with respect to an axis defined by the momentum of the positron. The azimuthal angle Φ is generated isotropically and Θ is computed from the energy-momentum conservation. Using this information enables the momentum vector of both photons to be calculated and transformed into the GEANT coordinate system.

```
CALL GANNIR
```

The routine GANNIR treates the special case when a positron falls below cutoff momentum (CUTELE in COMMON GCCUTS) before annihilating. In this case it is assumed that it comes to rest before annihiliting (see the tracking routine GTELEC). then GANNIR generates the photons, each of them has an energy k=m and the angular distribution is isotropic with gammas going in opposite direction.

2. Restrictions

1) The annihilation processes producing one or three or more photons are ignored, because these processes are negligible compared to the two-photon annihilation (see Refs 2 and 3).

2) Calculating the process it is assumed that the atomic electron initially is free and at rest. This is a usual assumption used inn shower programs (see Ref. 2).

3) In the case of a compound or mixture target material, the effective atomic number (computed in GSMIXT) is used.

3. References

1) W. Heitler, The Quantum Theory of radiation, Clarendon Press, Oxford (1954).

2) R.L. Ford, W.R. Nelson, SLAC-210, UC-32 (1978).

3) H. Messel, D.F. Crawford, Electron-Photon Shower Distribution Function, Pergamon press (1970).

Author(s) : G.N. Patrick
Origin : GEANT2

Submitted: 30.03.82
Revised: 13.05.86

Simulation of Particle Decays in Flight

CALL GDECAY

GDECAY is a control routine for the direct simulation of particle decays in flight. From a reference list of parent particles it selects a 2 or 3 body decay mode using the known branching ratios and calls all the necessary routines needed to generate the vertex and secondary tracks. GDECAY is automatically called by the tracking routines.

Subroutines Called

GDECA2, GDECA3, GLOREN, GDROT

Input via commons GCTRAK, GCKINE
Output via common GCKING

1. Method

1) Upon entry to GDECAY a binary search is made through a reference list of parent particles. This list is stored in the JPART structure and currently contains the particles defined in GPART [CONS300]. If the current track does not correspond to one of these particles, control is returned ***without*** any decay generation. Up to six decay modes and their corresponding branching ratios are then extracted from the JPART data banks. See Table 1 in CONS 310.

2) Using the branching ratios a decay is randomly selected. For same particles the sum of the defined branching ratios may not equal 100%, in which case it is possible that ***no*** decay is selected.

3) Depending on whether a 2 or 3 body decay is selected, either GDECA2 or GDECA3 is called to generate the four−momenta of the secondary decay products isotopically in the CMS.

4) The secondary momentum vectors are finally transformed into the LAB system and rotated back into the GEANT co−ordinate frame. The kinematics of the NGKINE products is stored in the COMMON/GCKING/.

5) When a particle decays and no branching ratio are defined, then GDECAY calls the user routine GUDCAY.

GEANT 3.10 USER'S GUIDE PHYS 410

Author(s) : M. Hansroul
Origin : GEANT2

Submitted: 01.09.76
Revised: 27.05.86

Lorentz Transformation

```
CALL GLOREN(BETA,PA,PB*)
```

Transforms momentum and energy from one Lorentz frame (A) to another (B).

Input:

BETA(1)	components of velocity of frame B
BETA(2)	as seen from frame A
BETA(3)	
BETA(4)	1./sqrt(1. − beta**2)
PA(1)	
PA(2)	momentum components in frame A
PA(3)	
PA(4)	total energy in frame A

Output:

PB(1)	
PB(2)	same quantities in frame B
PB(3)	
PB(4)	

$$PB(4) = \gamma[PA(4) - \vec{\beta} \bullet \vec{P}A]$$

$$\vec{P}B = \vec{P}A + \vec{\beta}\gamma\,[\gamma\vec{\beta} \bullet \vec{P}A/(\gamma+1) - PA(4)]$$

Author(s) : G.N.Patrick, L.Urban, D.Ward
Origin : GEANT2

Submitted: 12.03.82
Revised: 29.05.86

Ionisation Processes Induced by Heavy Particles

Lets call [$d\sigma(E,T)/dT$] the differential cross-section for the ejection of an electron of kinetic energy T by a charged particle of total energy E moving in a medium of density ρ.

TCUT is the kinematic energy cutoff below which the soft emitted electrons are treated as continuous energy lost by the incident particle, and above which they are explicitly generated (DCUTM in the program).

Then the mean value of the energy lost by the incident particle due to the ejected soft electrons is:

$$EL(E,TCUT) = \int_0^{TCUT} T \bullet \frac{d\sigma(E,T)}{dT} dT \qquad [1]$$

whereas the total cross-section for the ejection of an electron of energy T > TCUT is:

$$\sigma(E,TCUT) = \int_{TCUT}^{TMAX} \frac{d\sigma(E,T)}{dT} dT \qquad [2]$$

where TMAX is the maximum energy transferable to the free electron:

$$TMAX = \frac{2m(\gamma^2 - 1)}{1 + 2\gamma(m/M) + (m/M)^2} \qquad [3]$$

m = electron mass
M = mass of the incident particle
γ = E/M
β = P/M

Subroutine GDRELP

The integration of [1] leads to the restricted energy loss formula [1]:

$$\frac{1}{\rho}\left(\frac{DE}{DX}\right) = D\frac{Z}{A}\frac{1}{\beta^2}\left[\frac{1}{2}\ln\left(\frac{TMAX \bullet TC}{I^2}\right) - \frac{\beta^2}{2} - \frac{\delta}{2} - \frac{C_e}{Z}\right] \qquad [4]$$

where, $D = 4\pi N_A r_e^2 mc^2 = 0.000307$ GeV cm²/g
Z = atomic number of medium

$A =$ atomic weight
$\rho =$ density of medium
$TC =$ MIN(TCUT,TMAX)

I is the average ionisation potential of the atom in question. There exists a variety of phenomenological approximations for this, and the following quoted by Bricman et al. (2) has been adopted here

$$I = 16(Z)^{0.9} \text{ eV} \qquad [5]$$

This somewhat arbitrary choice does not, however, represent a serious source of error since I only enters into the logarithmic term of the equation.

δ is a correction term which takes account of the reduction in energy loss due to the so-called density effect. This becomes important at high energy because media have a tendency to become polarised as the incident particle velocity increases. As a consequence, the atoms in a medium can no longer be considered as isolated. To correct for this effect the formulation of Sternheimer (3) is used:

$$\delta \begin{cases} = 0 & X < X_0 \\ = 4.606X + C + a(X_1 - X)^m & X_0 < X < X_1 \\ = 4.606X + C & X \geq X_1 \end{cases} \qquad [6]$$

where, $4.606X = \ln(\gamma^2 - 1)$

X_0, X_1, a, m and C are constants which depend on the medium and are calculated as below:

$$C = -2\ln(I/h\nu_p) - 1$$

where, ν_p = plasma frequency of medium $= (ne^2/\pi m)\ s^{-1}$
n = number electrons / cm^3

$$a = \frac{4.606(X_a - X_0)}{(X_1 - X_0)^m}$$

where, $X_a = -C/4.606$

$I <$ 100 eV: $X_1 = 2.0, m = 3.0$
i) $X_0 = 0.2$ for $-C < 3.681$
ii) $X_0 = -0.326C - 1.0$ for $-C > 3.681$

$I \geq$ 100 eV: $X_1 = 3.0, m = 3.0$
i) $X_0 = 0.2$ for $-C < 5.215$
ii) $X_0 = -0.326C - 1.5$ for $-C > 5.215$

C_e/Z is a shell-correction term which accounts for the fact that at low energies (light elements) or for heavy elements (all energies) the probability of particle-electron collisions at deep electronic layers (K, L, etc.) is negligible. Barkas (4) has published a semi-empirical formula which is applicable to all materials, and this is utilised:

$$C_e(I,\eta) = (0.42237\eta^{-2} + 0.304\eta^{-4} - 0.00038\eta^{-6})10^{-6}I^2 + (3.858\eta^{-2} - 0.1668\eta^{-4} + 0.00158\eta^{-6})10^{-9}I^3$$

where, the units are MeV if I is expressed in eV ! and $\eta = \beta.\gamma$

However, this formula breaks down at very low energies and is thus only applied if $\eta > 0.13$.

The routine GDRELP(A,Z,DENS,T,DEDX) compute the formula [4] above (in GeV . cm^2/g) for an **incident proton** of kinetic energy T, in a medium of atomic weight A, atomic number Z, density DENS. (The density is just used for the calculation of the density effect δ).

GDRELP has been tested against energy loss tables (5,6) for various materials in the energy range 0 - 25 GeV. Typical discrepancies are as follows:

Beryllium	: 1.1 % at 0.05 GeV	0.02 % at 25 GeV
Hydrogen	: 1.5 % at 0.05 GeV	12.1 % at 25 GeV
Water	: 8.1 % at 0.05 GeV	4.4 % at 6 GeV

Subroutine GDRELA

The energy lost due to the soft delta rays is tabulated at initialisation time as a function of the medium and of the energy by routine GDRELA (see JMATE structure). The energy binning is set within the array ELOW (common GCMULO) in the routine GPHYSI.

The tables is filled with the quantity dE/dX in GeV/cm (formula 4 above) and for a molecule or a mixture:

$$\frac{dE}{dX} = \rho \sum_i W_i \frac{1}{\rho_i} \left(\frac{dE}{dX}\right)_i \qquad [7]$$

where W_i is the proportion by weight of the i-th element.

Description of banks

In fact 2 tables are filled from the formula [4] above:

- for muon, where the dE/dX due to the ionisation is summed up with the energy lost coming from others processes, namely Bremsstrahlung, pair production, nuclear interaction.
- for proton, where the dE/dX due to the ionisation is computed and stored alone. This table can be used for all charged particles other than electrons and muons by calculating the equivalent proton kinetic energy of the particle and searching for the correct energy bin in the table:

$$\text{Tproton} = \frac{\text{Mp}}{\text{M}} \text{T} \qquad [8]$$

T = kinetic energy of the projectile
M = mass of the projectile
Mp = proton mass

The pointers to the banks are: (see also JMATE data structure)

JMA = LQ(JMATE−I)	pointer to the I-th material
JEL1 = LQ(JMA−1)	pointer for dE/dX for e^+ /e^-
JEL2 = LQ(JMA−2)	pointer for dE/dX for μ^+ /μ^-
JEL3 = LQ(JMA−3)	pointer for dE/dX for all others particles

Subroutine GDRSGA

The integration of formula [2] gives the total cross-section :

$$\sigma(Z,E,TCUT) = 2\pi r_0{}^2 m \bullet \frac{Z}{\beta^2} \; \frac{1}{TCUT}(1-Y+\beta^2 Y \ln Y) \text{ for spin 0 particle} \qquad [9a]$$

$$\sigma(Z,E,TCUT) = \sigma(Z,E,TCUT) + 2\pi r^2{}_0 m \; \frac{Z}{\beta^2} \left(\frac{TMAX-TCUT}{2E^2}\right) \text{ for spin } {}^1/_2 \text{ particle} \qquad [9b]$$

where $Y = TCUT/TMAX \leq 1$

The macroscopic cross-section Σ (in 1/cm) is:

$$\Sigma = \frac{N\rho\sigma(Z,E,TCUT)}{A}$$

$$\Sigma = \frac{N\rho\sum_i p_i\sigma(Z_i,E,TCUT)}{\sum_i p_i A_i} = N\rho \sum_i (w_i/A_i) \bullet \sigma(Z_i,E,TCUT) \qquad [10]$$

for the chemical element or compound/mixture respectively where,

N Avogadro's number
$Z(Z_i)$ atomic number (of i-th component) of the medium
$A(A_i)$ atomic number (of i-th component) of the medium
ρ density
σ total cross-section
p_i proportion by number of the i-th element in the material $p_i \simeq w_i/A_i$ where w_i is the corresponding proportion by weight
TCUT electron cut-off energy (DCUTM in the program).

The mean free path, $\lambda = 1/\Sigma$ (in cm) is tabulated at initialisation time as a function of the medium and of the energy by routine GDRSGA **for muon only.** The energy binning is set within the array ELOW (common GCMULO) in the routine GPHYSI. (See JMATE data structure).

JMA = LQ(JMATE−I)	pointer to the I-th material
JDRAY = LQ(JMA−11)	pointer to Delta rays cross-section bank
JDRAY	pointer for electrons
JDRAY+90	pointer for positrons
JDRAY+180	pointer for μ^+ $/\mu^-$

The cross-section [9] is strongly dependent of the mass of the incident particle, and cannot be tabulated in a general way for any charged handrons. Therefore, for such particles, the cross-section [10] is computed at tracking time within the routine GTHADR.

1. References

1) Review of Particle Properties (Particle Data Group) — April 84, Edition (page S50)

2) C. Bricman et al. : — Rev. Mod. Phys. 52 (1980) 1

3) R.M. Sternheimer : — Phys. Rev. 88 (1952) 851
Phys. Rev. 103 (1956) 511

4) W.H. Barkas : — UCRL - 10292 (August 1962)

5) C. Serre : — CERN 67-5 (March 1967)

6) C. Richard - Serre : — CERN 71-18 (Sept. 1971)

Author(s) : L.Urban
Origin : Same

Submitted: 26.10.84
Revised: 27.04.87

Total cross-section and energy loss for Bremsstrahlung by Muons

1. Method

Let us call [$d\sigma(Z,T,k)/dk$] the differential cross-section for production of a photon of energy k by a muon of kinetic energy T in the field of an atom of charge Z.

k_c is the energy cutoff below which the soft photons are treated as a continuous energy lost by the muon, and above which they are explicitly generated **(BCUTM in the program.)**

Then the mean value of the energy lost by the muon due to the soft photons is:

$$EL(Z,T,k_c) = \int_0^{k_c} k \bullet \frac{d\sigma(Z,T,k)}{dk}\, dk \qquad [1]$$

whereas the total cross-section for the emission of a photon of energy > k_c

$$\sigma(Z,T,k_c) = \int_{k_c}^{T} \frac{d\sigma(Z,T,k)}{dk}\, dk \qquad [2]$$

The most accurate cross section formula for the high energy ($T \geq 1GeV$) muon Bremsstrahlung can be found in a paper of W. Lohmann, R. Voss (CERN Yellow Report 85 03, 1985). We have used this cross section to calculate "data" points $\sigma(Z,T,k_c)$ and $EL(Z,T,k_c)$. We have parametrized the cross section and energy loss as:

$$\sigma(Z,T,k_c) = Z\,[Z + \xi_\sigma\,(1 + \gamma \ln Z)\,]\left[\ln\frac{k_{max}}{k_c}\right]^{\alpha} F_\sigma\,(Z,X,Y) \qquad \text{(in barn)} \quad [3]$$

$$EL(Z,T,k_c) = Z\,[Z + \xi_l\,(1 + \delta \ln Z)\,]\left[\ln\frac{k_c}{E}\right]^{\beta} F_l\,(Z,X,Y) \qquad \text{(in GeV barn)} \quad [4]$$

where $k_{max} = E - 0.75\sqrt{e}\; m_\mu Z^{1/3}$ is the maximum possible value of the photon energy,

$X = \ln(E/m_\mu)$

$Y = \ln(k_c/m_\mu)$

$E = T + m_\mu$

$e = 2.7182....$

The functions $F_i(Z,X,Y)$ $(i=\sigma,l)$ are polynomials of their variables

$$\begin{aligned} F_i(Z,X,Y) \quad &= (C_1+C_2X+\ldots+C_6X^5)+(C_7+C_8X+\ldots+C_{12}X^5)Y \\ &+\ldots.+(C_{31}+C_{32}X+\ldots+C_{36}X^5)Y^5 \\ &+Z[(C_{37}+C_{38}X+\ldots+C_{40}X^3)+(C_{41}+C_{42}X+\ldots+C_{44}X^3)Y \\ &+\ldots+(C_{48}+C_{49}X+\ldots+C_{52}X^3)Y^3)] \end{aligned} \qquad [5]$$

We have computed more than 2000 "data" points in the range:

$Z = 1,6,13,26,50,82,92$

$1\mathrm{GeV} \le T \le 10\mathrm{TeV}$

$10\mathrm{KeV} \le k_c \le T$

and performed a least squares fit to determine the values of the parameters in the formulae [3] to [5]. These values ($\xi\sigma$, γ, α, C_i for σ and ξ_l, δ, β, C_i for EL) can be found in DATA statements within the functions GBRSGM and GBRELM which compute the formulae [3] and [4] respectively.

The accuracy of the fit can be estimated as:

$$\frac{\Delta\sigma}{\sigma} \le \begin{array}{ll} 10-12\% & \text{for } T\le 5\mathrm{GeV} \\ 4\% & \text{for } T > 5\mathrm{GeV} \end{array}$$

$$\frac{\Delta EL}{EL} \le \begin{array}{ll} 10\% & \text{for } T\le 5\mathrm{GeV} \\ 3\% & \text{for } T > 5\mathrm{GeV} \end{array}$$

(But the contribution of the Bremsstrahlung to the total energy loss of the muons is less than 1% for $T \le 5\mathrm{GeV}$.)

We have made a separate parametrisation for the total energy lost by the muon due to Bremsstrahlung. If $k_c \ge k_{max}$. Then we have chosen the the following parametrisation:

$$EL^{brem}(Z,T) = EL(Z,T,k=k_{max}) =$$

$$Z(Z+1)\,k_{max}[d_1+(d_2X+d_3Y) \qquad [6]$$

$$+(d_4X^2+d_5XY+d_6Y^2)+\ldots+(d_{22}X^6+d_{23}X^5Y+\ldots+d_{28}Y^6)]$$

$$k_{max} = E-0.75\sqrt{e}\,m_\mu Z^{1/3}$$

$$X = \ln(E/m_p)$$

$$Y = Z^{1/3}$$

$$E = T + m_\mu$$

The accuracy of the formula [6] is rather good,

$$\frac{\Delta EL^{brem}}{EL^{brem}} \leq 1.5\% \text{ if } T > 1\text{GeV} ;\ 1 \leq Z \leq 100$$

and

$$\frac{\Delta EL^{brem}}{EL^{brem}} \leq \qquad 1\% \quad \text{if } T > 5\text{GeV}$$

CALL GBRELA

The energy loss due to soft photon Bremsstrahlung is tabulated at initialisation time as a function of the medium and of the energy by routine GBRELA (see JMATE data structure). The energy binning is set within the array ELOW (COMMON CGMULO) in the routine GPHYSI.

$$\frac{dE}{dX} = \frac{N\rho}{A} \bullet EL(Z,T,k_c) \text{ in GeV/cm} \qquad [7]$$

N Avogadro's number
A atomic weight of the material
ρ density of the material
EL formulae [4] above.

For a molecule or a mixture:

$$\frac{dE}{dX} = \rho \bullet \Sigma\ w_i \frac{1}{\rho_i} \bullet (dE/dX)_i \qquad [8]$$

w_i proportion by weight of the i-th element.

In the Tables, the dE/dX due to Bremsstrahlung is summed up with the energy lost coming from ionisation, pair production, nuclear interaction.

JMA = LQ(JMATE−I)	pointer to the I-th material
JEL2 = LQ(JMA−2)	pointer to dE/dX for μ^+ /μ^-

CALL GBRSGA

The mean free path, λ, for an electron to radiate a photon via Bremsstrahlung is given by

$$\lambda = 1/\Sigma \qquad [9]$$

where Σ is the macroscopic cross-section (1/cm). This macroscopic cross-section can be written as

$$\Sigma = \frac{N\rho\sigma(Z,E,k_c)}{A} \qquad [10a]$$

for a compound or mixture:

$$\Sigma = \frac{N\rho \sum_i p_i \sigma(Z_i,E,k_c)}{\sum_i p_i A_i} = N\rho \sum_i (w_i/A_i) \bullet \sigma(Z_i,E,k_c) \qquad [10b]$$

N	Avogadro's number
$Z(Z_i)$	atomic number of the material (i-th component of the material)
$A(A_i)$	atomic weight of the material (i-th component)
ρ	density of the material
σ	total cross-section per atom for discrete (photon energy = k_c) Bremsstrahlung (formula [3] above)
p_i	proportion by number of the i-th element in the material $p_i \simeq w_i/A_i$ where w_i is the corresponding proportion by weight
k_c	photon cut-off energy (BCUTM in the program).

This mean free path is tabuled at initialisation time as a function of the medium and of the energy by routine GBRSGA (see JMATE data structure). The energy binning is set within the array ELOW (COMMON GCMULO) in the routine GPHYSI.

JMA = LQ(JMATE−I)	pointer to the I-th material
JBREM = LQ(JMA−9)	pointer to Brems cross-sections
JBREM	pointer for e^+ /e^-
JBREM+90	pointer for μ^+ /μ^-

Author(s) : L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 30.05.86

Simulation of Discrete Bremsstrahlung by Muons

CALL GBREMM

GBREMM generates Photon Bremsstralung by high energetic muon by treating it as a discrete process.

Input : via COMMON /GCTRAK/
Output: via COMMON /GCKING/

GBREMM is automaticaly called from the tracking routines if and when the parent muon reaches its radiation point during the tracking stage of GEANT.

1. Method

The differential cross-section of a photon of energy K by a muon of energy E is:

$$\frac{d\sigma}{dv} = N \bullet \frac{1}{v}\left(\frac{4}{3} - \frac{4}{3}v+v^2\right) \bullet \Phi(\delta) \qquad [1]$$

where, N = normalisation factor (not important here)

$v = k/E$

$$\delta = \frac{M^2}{2E} \bullet \frac{v}{1-v}$$ is the minimum momentum transfer to the nucleous

M = muon mass
m = electron mass
e = 2.718

$$\Phi(\delta) = \ln\left(\frac{189M}{mZ^{1/3}}\right) - \ln\left(\frac{189\sqrt{e}}{mZ^{1/3}} \bullet \delta + 1\right) \text{ if } Z \leq 10$$

$$\Phi(\delta) = \Phi(\delta) + \ln\left(\frac{2}{3Z^{1/3}}\right) \text{ if } Z > 10$$

and

$$v_c = \frac{kc}{E} \leq v \leq \left(1 - \frac{3\sqrt{e}MZ^{1/3}}{4E}\right) = V_{MAX}$$

Therefore, the differential cross-section can be written as

$$\frac{d\sigma}{dv} \simeq f(v) \bullet g(v) \qquad [2]$$

with

$$f(v) = \left[v \bullet \ln \left(\frac{V_{MAX}}{vc} \right) \right]^{-1}$$

$$g(v) = \frac{1}{\Phi(0)} \bullet \left(1 - v + \frac{3v^2}{4} \right) \bullet \Phi(\delta)$$

we can sample the photon energy in the following way:

R1, R2 are iniformly distributed random numbers [0,1]

step 1 compute $v = v_c \bullet (V_{MAX}/V_c)^{R1}$

step 2 use v to compute the rejection function g(v) and:
if R2 > g(v) go back to step 1

step 3 $k = v \bullet E$

After the successful sampling of k, GBREMM generates the polar angles of the radiated photon with respect to an axis defined along the parent muon's momentum. Φ is generated isotropically and Θ is assigned the approximate average value [2]:

$$\Theta = M/E$$

2. References

1) W. Lohmann, R. Kopp, R. Voss, CERN, Yellow Report, 85-03.

2) P. Marmier, E. Sheldon, Physics of Nuclei and Particles, Vol. 1, Academic Press, (1969) 140.

Author(s) : L.Urban
Origin : Same

Submitted: 24.03.86
Revised: 21.05.87

Total cross-section and energy loss for e^+ e^- production by Muons

1. Principle

When a muon of total (high) energy E is moving in the field of an atom of charge Z, it can radiate a pair (e^+ ,e^-) (4th order QED process) Lets call E^+ and E^- the total energy of the emitted positron and electron respectively.

Defining:

$$v = \frac{E^+ + E^-}{E} \text{ and } \rho = \frac{E^+ - E^-}{E^+ + E^-} \qquad [1]$$

the differential cross-section for the process can be written [Ref. 1]:

$$\frac{d^2\sigma}{dv d\rho} = \alpha^4 \bullet (Z\lambda\!\!\!^-)^2 \bullet [\phi_e + (m/M)^2 \phi_\mu] \qquad [2]$$

where

α = 1./137 is the fine structure constant
$\lambda\!\!\!^-$ = 3.8616 10^{-11} cm in the electron compton wavelenght
m = electron mass
M = muon mass
T = E−M is the kinetic energy of the muon.

The explicit form of the terms ϕ_e and ϕ_μ can be found in Ref. 1

The kinematic range of v and ρ are:

$$v_{\text{min}} = \frac{4m}{E} \leq v \leq v_{\text{max}} = \left(1 - 3\sqrt{e}\frac{M}{4E}Z^{1/3}\right) \qquad [3]$$

$$0 \leq |\rho(v)| \leq \rho_{\text{max}}(v) = \left[1 - \frac{6M^2}{E^2(1-v)}\right]\left[1 - \frac{4m}{vE}\right]^{1/2}$$

where e = 2.718

E_c is the total energy cutoff below which the emitted (e^+ ,e^-) are treated as continuous energy lost by the muon, and above which they are explicit generated (PPCUTM in the program), and

$$v_c = E_c/E \qquad [5]$$

Then the mean value of the energy lost by the incident muon due to the (soft) e^+ ,e^- pair is:

$$EL(Z,T,E_c) = 2E\int_{v_{\min}}^{v_c} dv\, v \int_0^{\rho_{\max}(v)} d\rho \frac{d^2\sigma}{dv d\rho} \qquad \text{(GeV barn / atom)} \quad [6]$$

whereas the total cross-section for the emission of

$$\sigma(Z,T,E_c) = 2\int_{v_c}^{v_{\max}} dv\, v \int_0^{\rho_{\max}(v)} d\rho \frac{d^2\sigma}{dv d\rho} \qquad \text{(Barn / atom)} \quad [7]$$

Instead of using the explicit formulae for $d^2\sigma/dvd\rho$ [Ref. 1] we have chosen to parametrize directly $EL(Z,T,E_c)$ and $\sigma(Z,T,E_c)$ as:

$$\sigma(Z,T,E_c) = Z[Z+\xi_s(1+\gamma lnZ)]\left[\ln\frac{E_{\max}}{E}\right]^{\alpha} F_\sigma(Z,X,Y) \qquad [8]$$

$$EL(Z,T,E_c) = Z[Z+\xi_l(1+\delta lnZ)] \bullet E \bullet \left[\frac{E_c - E_c^{\min}}{E}\right]^{\beta} F_l(Z,X,Y) \qquad [9]$$

where $\xi\sigma, 1$; $\alpha,\beta,\gamma,\delta$ are parameters to be fitted, $E_c^{\min} = E*V_{\min} = 4m_e$, $E_{\max} = E*V_{\max}$

The functions $F_i(Z,X,Y)$ $(i=\sigma,l)$ have the form:

$$F_i(Z,X,Y) = F_{i_0}(X,Y) + ZF_{i_1}(X,Y) \qquad [10]$$

where $F_{ij}(X,Y)$ denotes a function constructed from two polynomials

$$F_{ij}(X,Y) = \left| \begin{array}{ll} P_{ij}^{neg}(X,Y) & \text{if } Y \le 0 \\ P_{ij}^{pos}(X,Y) & \text{if } Y > 0 \end{array} \right. \qquad [11]$$

and P_{ij}^{neg}, P_{ij}^{pos} fulfill the conditions:

$$P_{ij}^{neg}(X,Y=0) = P_{ij}^{pos}(X,Y=0) \qquad [12]$$

$$\frac{\delta P_{ij}}{\delta Y\ |Y=0} = \frac{\delta P_{ij}^{pos}}{\delta Y\ |Y=0} \qquad [13]$$

The detailed form of the P_{ij} polynomials are:

$$P_{i_0}^{neg}(X,Y) = (C_1+C_2X+\ldots+C_6X^5)+(C_7+C_8X+\ldots+C_{12}X^5)Y$$
$$(C_{13}+C_{14}X+\ldots+C_{18}X^5)Y^2+\ldots+(C_{31}+\ldots+C_{36}X^5)Y^5$$
$$P_{i_0}^{pos}(X,Y) = (C_1+C_2X+\ldots+C_6X^5)+(C_7+C_8X+\ldots+C_{12}X^5)Y$$
$$+(C_{37}+C_{38}X+\ldots+C_{42}X^5)Y^2+\ldots+(C_{55}+C_{56}X+\ldots+C_{60}X^5)Y^5 \qquad [14]$$
$$P_{i_1}^{neg}(X,Y) = (C_{61}+C_{62}X+\ldots+C_{65}X^4)+(C_{66}+C_{67}X+\ldots+C_{70}X^4)Y$$
$$+(C_{71}+C_{72}X+\ldots+C_{75}X^4)Y^2+\ldots+(C_{81}+C_{82}X+\ldots+C_{85}X^4)Y^4$$
$$P_{i_1}^{pos}(X,Y) = (C_{61}+C_{62}X+\ldots+C_{65}X^4)+(C_{66}+C_{67}X+\ldots+C_{70}X^4)Y$$
$$+(C_{86}+C_{87}X+\ldots+C_{90}X^4)Y^2+\ldots+(C_{96}+C_{97}X+\ldots+C_{100}X^4)Y^4$$

with

$X=\ln(E/M)$

$Y=\ln(E_c/v_jE)$ for $j=\sigma,1$

(v_s, v_l parameters to be fixed).

By a numerical (twofold) integration of the formulae [2], [6], [7] above, we have calculated 1800 "data points" in the range

$$\begin{aligned} &Z = 1,6,13,26,50,82,92 \\ &1\text{GeV} \le T \le 10\text{ TeV} \\ &4m \le E_c \le T \end{aligned}$$

and performed a least squares fit to determine the parameters. The fitted values of the parameters are in DATA statements in the functions GPRSGM and GPRELM, which compute the formulae [9] and [8] respectively. The accuracy of the fit:

$$\frac{\Delta\sigma}{\sigma} \le \begin{array}{ll} 10\% & \text{for } T\le 5\text{GeV} \\ 5\% & \text{for } T > 5\text{GeV} \end{array}$$

$$\frac{\Delta EL}{EL} \le \begin{array}{ll} 10\% & \text{for } T\le 5\text{GeV} \\ 4\% & \text{for } T > 5\text{GeV} \end{array}$$

The function GPRELM contains a second formula to calculate the total energy lost by the muon due to direct e^+ e^- production (this formula is used if $E_c \ge E_{max} = E-0.75\sqrt{e}\, m_\mu Z^{1/3}$). This formula gives the total energy loss with an error less than 1%. It has the form:

$$EL^{pair}(Z,T) = EL(Z,T,E_c=E_{max})$$
$$= Z(Z+1)\, E_{max}\, [d_1+(d_2X+d_3Y) \qquad [15]$$
$$+ (d_4X^2+d_5XY^2)+\ldots$$
$$(d_{22}X^6+d_{23}X^5Y+\ldots+d_{28}Y^6)]$$

where $X=\ln(E/M)$; $X=Z^{1/3}$

the fitted parameters d_i can be found in DATA within the function GPRELM.

CALL GPRELA

The energy lost by the muon due to the (soft) $e^+ e^-$ pair radiated is tabulated at initialisation time as a function of the medium and of the energy by the routine GPRELA (see JMATE data structure). The energy binning is set within the array ELOW (COMMON CGMULO) in the routine GPHYSI.

$$\frac{dE}{dX} = \frac{N\rho}{A} \bullet EL(Z,T,E_c) \quad \text{in GeV/cm} \qquad [16]$$

N	Avogadro's number
A	atomic weight of the material
ρ	density of the material (in g/cm^3)
EL	formulae [9] above.

For a mixture or a molecule:

$$\frac{dE}{dX} = \rho \bullet \Sigma\ w_i \frac{1}{\rho_i} \bullet (dE/dX)_i \qquad [17]$$

w_i proportion by weight of the i-th element.

In the Table, the dE/dX due to the $e^+ e^-$ pair production is summed up wit the energy loss coming from others processes, namely: ionisation, Bremsstrahlung, nuclear interaction.

JMA = LQ(JMATE − I)	pointer to the I-th material
JEL2 = LQ(JMA − 2)	pointer to dE/dX for μ^+ / μ^-

CALL GPRSGA

The mean free path, λ, for a muon to radiate a $e^+ e^-$ pair is given by

$$\lambda = 1/\Sigma \ \text{(in cm)} \qquad [18]$$

where Σ is the macroscopic cross-section (1/cm). This macroscopic cross-section can be written as

$$\Sigma = \frac{N\rho\sigma(Z,E,E_c)}{A} \qquad [19a]$$

for a compound or mixture:

$$\Sigma = \frac{N\rho\sum_i p_i\sigma(Z_i,E,E_c)}{\sum_i p_i A_i} \qquad [19b]$$

N Avogadro's number
$Z(Z_i)$ atomic number of the material (i-th component of the material)
$A(A_i)$ atomic weight of the material (i-th component)
ρ density of the material
σ total cross-section per atom for pair production (formula [8] above)
p_i proportion by number of the i-th element in the material $p_i \simeq w_i/A_i$ where w_i is the corresponding proportion by weight
E_c cut-off energy (PPCUTM in the program).

This mean free path is tabulated at initialisation time as a function of the medium and of the energy by routine GPRSGA (see JMATE data structure). The energy binning is set within the array ELOW (COMMON GCMULO) in the routine GPHYSI.

Pointers

JMA = LQ(JMATE−I)	pointer to the I-th material
JPAIR = LQ(JMA−10)	pointer to pair cross-sections
JPAIR+90	pointer for muons

2. Reference

1. W. Lohmann et al., CERN Yellow report 85-03.

Author(s) : L. Urban
Origin : Same

Submitted: 26.10.84
Revised: 28.05.86

Simulation of e^+ e^- production by Muons

CALL GPAIRM

GPAIRM generates direct e^+ e^- pair radiated by high energetic muon.

Input : via COMMON /GCTRAK/
Output: via COMMON /GCKING/

GPAIRM is automaticaly called from the tracking routine GTMUON if and when the parent muon reaches its radiation point during the tracking stage of GEANT.

1. Method

The double differential cross-section for the process can be written (Ref. 1):

$$\frac{d^2\sigma}{dvd\rho} = \alpha^4 \quad \frac{2}{3\pi}(Z\lambda)^2 \bullet \frac{1-v}{v}\left[\phi_e + (m/M)^2\,\phi_\mu\right] \qquad [1]$$

All the quantities in the expression above are defined in PHYS 450.

By computing this cross-section for differents (v,ρ) points, it can be seen:

1) the **shape** of the functions:

$$\frac{d^2\sigma}{dvd\rho}$$

and

$$\frac{d\sigma}{dv} = \int d\rho \quad \frac{d^2\sigma}{dvd\rho}$$

pratically does not depend on Z

2) the dominant contribution comes from **the low v region:**

$$v_{min} = (4m/E) \leq v \leq 100^* v_{min} \qquad [2]$$

3) in this low region ($d^2\sigma/dvd\rho$) **is flat** as a function of ρ

Therefore, we propose the following sampling method as a rough approximation.

{1} In the low v region the differential cross-section

$$\frac{d\sigma}{dv} = \int d\rho \quad \frac{d^2\sigma}{dv d\rho}$$

can be approximated as:

$$\frac{d\sigma}{dv} \simeq \left[1 - \frac{v_{min}}{v}\right]^{1/2} \frac{1}{v^a} \qquad [3]$$

where $a = 2-(\ln E/10)$ (E in GeV) [4]

we can write:

$$\frac{d\sigma}{dv} \simeq f(v) \bullet g(v) \qquad [5]$$

where,

$$f(v) = \frac{(a-1)}{(1/v_c^{a-1}) - (1/v_{max})^{a-1}} \quad \frac{1}{v^a} \qquad [6]$$

normalized distribution in $[v_c, v_{max}]$

$$g(v) = \left[1 - \frac{v_{min}}{v}\right]^{1/2} \text{ rejection function} \qquad [7]$$

{2} R1 and R2 being two uniformly distributed random numbers in [0,1]:

- Sample v from the distribution f(v) as:

$$v = \left(\frac{1-R1}{v_c^{a-1}} - \frac{R1}{v_{m\ ax}^{a-1}}\right)^{1/(1-a)} \qquad [8]$$

- Accept v if $R2 \leq g(v)$ [9]

{3} Then compute

$$\rho_{max}(v) = \left[1 - \frac{6M^2}{E^2(1-v)}\right] \left[1 - \frac{4m}{vE}\right]^{1/2} \qquad [10]$$

and generate ρ uniformly in $[-\rho_{max}, +\rho_{max}]$ range.

After the successful sampling of (v,ρ), GPAIRM generates the polar angles of the $(e^+ \; e^-)$ radiated pair with respect to an axis defined along the parent muon's momentum. Θ is assigned the approximate average value:

$$\Theta = M/E \qquad [11]$$

ϕ^+ is generated isotropically and $\phi^- = \phi^+ + \pi$

2. References

1) W. Lohmann, R. Kopp, R. Voss, CERN, Yellow Report, 85-03.

GEANT 3.10 USER'S GUIDE PHYS 460

Author(s) : F. Carminati
Origin : H.C. Fesefeldt

Submitted: 20.12.85
Revised: 13.05.86

Muon-nucleus Interactions

1. Introduction

This set of routines generates the interactions of muons with the nuclei of the tracking material. The code is a straight translation into the GEANT dialect of the corresponding code from the GHEISHA Montecarlo Program. The GHEISHA routines CASMU and CALIM have been used. *

The information contained in this chapter is mainly taken from the Gheisha manual (see note) to which the user is referred.

The muon-nucleus inelastic cross-section is taken as 0.0003 mb for nucleon when the energy of the incoming muon is $E < 30$ GeV, and is slowly increasing for $E > 30$ GeV according to the law:

$$\sigma = 0.3 * (E/30)^{0.25} \quad [\mu b]$$

The energy and angle of final state muon is generated according to the 'free quark parton model'. If E is the energy of the incoming muon and Ω and W the angle and the energy of the outgoing muon, the differential cross-section can be written as:

$$\frac{d\sigma}{d\Omega dW} = \Gamma\,(\sigma_T + \epsilon\sigma_S)$$

where:

$$\Gamma = \frac{k\alpha}{2\pi^2 Q^2}\ \frac{W}{E}\ \frac{1}{1-\epsilon}$$

$$\epsilon = \left[\,1+2\ \frac{Q^2+\nu^2}{Q^2}\ \tan^2\frac{\theta}{2}\,\right]$$

Q^2 and ν are the normal scaling variables expressed by:

$$Q^2 = -q^2 = 2(EW - |p||p'|\cos\theta - m_\mu^2) \qquad \nu = E - W$$

here σ_T and σ_S are the photoabsorbtion cross-sections for transverse and longitudinal photons respectively for which the relation is used:

*) GHEISHA: H.C. Fesefeldt III. Physikalisches Institut der RWTH Aachen Physikzentrum, 5100 Aachen, W. Germany, tel. 0241/807274) version 7.03, released on September, 28th, 1985.
For additional informations see "Simulation of Hadronic Showers, Physics and Applications", PITHA-report 85-02, H.C. Fesefeldt, RWTH Aachen.

$$\sigma_S = 0.3 \left(1 - \frac{1}{1.868} \quad \frac{Q^2}{\nu}\right) \sigma_T$$

and σ_T is assumed to be constant $\sigma_T = 0.12$ mb. For the incident flux K of the photons the Gillman's convention is used:

$$K = \nu + Q^2/2\nu$$

A three dimensional important sampling in the variables E, W and θ is performed each time an interaction has to occur.

The hadrons are generated in an approximate way. The virtual photon is replaced by a real pion of random charge with the same kinetic energy. Then the GUHADR routine is called to generate a pion-nucleus inelastic scattering. While the final state generated this way gives a good approximation for calorimetric purposes, the kinematic of the final state may be a rather poor approximation of reality.

The muon nucleus interactions are activated by the MUNU data card of GEANT. After a muon-nucleus interaction the muon will still be the current particle. If MUNU 1 has been specified, secondaries coming from the interaction of the virtual photon with the nucleus will be in the GEANT temporary stack. If MUNU 2 has been specified, then the secondary particles will not be generated and the energy lost by the muon will be added to DESTEP.

For each material a table of muon-nucleus cross-sections is stored at initialization time. See material bank structure for details.

2. Description of the routines

CALL GMUNUI

This routines computes and stores in the appropriate bank the value of the muon-nucleus cross-section for a given material. It is called by GPHYSI.

CALL GMUNU

This routine is called by GTMUON every time a muon-nucleus interaction has to happen. It generates the final state particles as well as the outgoing muon. A call to GUHADR is performed if IMUNU (which is the variable set by the MUNU data card) is equal to 1. If the GHEISHA interface is used, an inelastic interaction is forced (which could also be a fission in case of heavy materials). The secondaries from the π-nucleus interaction are always generated if IMUNU is equal to 1, irrespectively from the value of IHADR.

GMUSIG(E,E1,COSTET)

This routine returns the value of the differential cross-section in millibarns for a muon of energy E to generate a nuclear interaction and giving an outgoing muon of energy E1 at an angle the cosine of which is COSTET.

GEANT 3.10 USER'S GUIDE PHYS 500

Author(s) : R.Brun, F.Carena
Origin : T.Baroncelli

Submitted: 26.04.82
Revised: 02.05.86

The GEANT/TATINA interface

CALL GPHADR

Called at each step in order to compute the remaining distance to the hadronic interaction point (see PHYS 010, formula 4).

The mean free path λ is re evaluated at each step:

$$\lambda = \frac{1}{\Sigma} = \frac{A}{N.\rho.\sigma(Z,E)} \qquad [1]$$

where,

- N Avogadro's number
- Z,A atomic number and weight of the medium
- ρ density of the medium
- σ total cross-section per atom for hadronic interaction, computed by the function GPHSIGM below.

FUNCTION GIISIGM(P,IPART,A)

Returns absorption cross-section in millibarns for a particle with momentum P (GeV/C).

GEANT type IPART on a hadron of atomic number A.

The cross-section table has been provided by R. Barlow (Manchester), January 1984.

CALL GHTATI

The interface routine with the T. Baroncelli routines of TATINA.

Generate the finale particle's state after an hadronic interaction.

Author(s) : F. Carminati
Origin : H.C. Fesefeldt

Submitted: 24.02.86
Revised: 28.05.86

The GEANT/GHEISHA Interface

1. Introduction

This package contains an interface between GEANT and the hadronic shower development program GHEISHA*

This interface generates hadronic interactions with the nuclei of the current tracking medium according to the GHEISHA code (including cross-sections calculation and final state multiplicity and kinematics), while the GEANT philosophy is preserved for the tracking.

The GHEISHA code is stored in a separate pam file.

It is not possible to link directly the GHEISHA library, because some slight modifications are required. A special library to be linked to the GEANT user program must be provided. Decks to install this library are available in the GEANX310 pam file via the following PATCHY cradle:

```
+USE,INSTALL,GHEISHA,IBM.   or VAX, APOLLO, VMCMS, CDC and NORD
+PAM,11.                    here attach GEANX310 pam file
```

The GHEISHA printing flags are set via the SWIT card of GEANT with the following rule. Each switch greater than 100 but smaller that 111 sets the corresponding printing flag of GHEISHA module 100, so that SWIT 105 will set the printing flag 5 of GHEISHA. The printing flags of GHEISHA have the following meaning:

NPRT(1)	one header for each track in the shower
NPRT(2)	all tracking information
NPRT(3)	kinematic of decays (not effective)
NPRT(4)	kinematic of nuclear interactions
NPRT(5)	kinematic of electromagnetic interactions (not effective)
NPRT(6)	material constants, dE/dX and absorbed energies (not effective)
NPRT(7)	event summary
NPRT(8)	history of all interactions/decays
NPRT(9)	free
NPRT(10)	tables of the geometry, cross-sections, etc.

NPRT(1), NPRT(2) and NPRT(6) should be used only in case of errors and bugs. NPRT(8) produces the most illustrative output. Those flags work in conjunction with the DEBUG data card (and the IDEBUG switch) of GEANT.

*) GHEISHA: H.C. Fesefeldt III. Physikalisches Institut der RWTH Aachen Physikzentrum, 5100 Aachen, W. Germany, tel. 0241/807274) version 7.03, released on September, 28th, 1985.
For additional informations see "Simulation of hadronic showers", PITHA-report 85−02, H.C. Fesefeldt, RWTH Aachen.

2. Description of the routines

GHESIG(P,EK,AVER,A,Z,W,NLM,DENS,CORR,IPART)

This function returns the total macroscopic cross-section in cm^{-1}. The meaning of the parameters is:

P momentum of the particle (in GeV/c)
EK kinetic energy of the partcle (in GeV)
AVER average mass number of the material
A vector of length NLM containing the mass numbers of the components of the mixture, the same as AVER in case of a non compound material
Z vector of length NLM containing the atomic numbers of the components of the mixture, or the atomic number in case of a non compound material
W vector of length NLM containing the relative weights of the components of the mixture (normalized to one), one in case of a non compound material
NLM number of components of the mixture, one in case of a non compound material
DENS density of the material
CORR correction flag, if this flag is set, then corrections are applied to the cross-section in case of mixture (like BGO, see below for more details)
IPART GEANT particle code

The correction flag, if present, is stored in the 26th word of the second bank in the tracking medium linear structure pointed by LQ(JTMED−NUMED).

The cross-sections on nucleus are known only for pions and protons. The general law:

$$\sigma(A) = 1.25 * \sigma_{tot}(\text{Proton}) * A^{\alpha}$$

is used but it is valid only for momenta > 2 GeV. The parametrisation done gives only a behaviour averaged over momenta and particle types. For a detector with only a few materials it would be of course much better to use tables of the measured cross-sections. For H, Al, Cu and Pb the measured cross-sections are stored in a data statement.

The cross-section as stored are the GHEISHA 7 cross-section precalculated. As a starting point the measured cross-sections of pion, kaon, proton, antiproton and neutron over protons are used. The cross-sections tabulated are measured values taken from the CERN-HERA-compilations. The values for K0S/K0L are updated until July 1980. Strange baryon cross-section are calculated using a parametrisation in terms of quark-quark forward scattering amplitudes and optical theorem. The additive quark-quark scattering model is used. The quark scattering amplitude are defined as follow:

<PP ?PP >	=	<NN ?NN >	= P
<PN ?PN >	=	<NP ?NP >	= P
<PN ?NP >	=	<NP ?PN >	= 0
<PL ?PL >	=	<NL ?NL >	= P − S
<PLB?PLB>	=	<NLB?NLB>	= P − S
<PL ?LP >	=	<NL ?LN >	= 0
<PBN?PBN>	=	<NBP?NBP>	= P
<PBP?PBP>	=	<NBN?NBN>	= P + A
<PBP?NBN>	=	<NBN?PBP>	= A
<LL ?LL >	=		= $(P - S)^2/P$
<LBL?LBL>	=		= $(P + A)*(P + S)^2/P^2$
<PBP?LBL>	=	<NBN?LBL>	= $A*(P - S)/P$

and from these the following cross-sections are computed:

PI− P	= 6P + 2A
PI+ P	= 6P + A
K− P	= 6P + 2A − 3S
K0S P	= 6P + A/2− 3S
K0L P	= 6P + A/2− 3S
K0 P	= 6P − 3S = = > σ(K0) = σ(K+)
K0B P	= 6P + A − 3S = = > σ(K0B)= σ(K+)/4+σ(K0L)/3+5σ(K−)/12
K+ P	= 6P − 3S
P P	= 9P
PB P	= 9P + 5A
N P	= 9P

Then the following amplitudes are computed:

P	= σ(P,P)/9
A	= (σ(PB,P)−σ(P,P))/5
S	= 2*σ(P,P)/9−σ(K+,P)/3

and from here:

NB P	= 9P + 4A
L P	= 9P − 3S
LB P	= 9P + 2A − 3S
S− P	= 9P − 3S
S+ P	= 9P − 3S
S−B P	= 9P + 2A − 3S
S+B P	= 9P + 4A − 3S
X0 P	= 9P − 6S
X− P	= 9P − 6S
X0B P	= 9P + 2A − 6S
X−B P	= 9P + A − 6S

All the cross-sections are contained in data statements so no external file is needed.

CALL GPGHEI

This routine returns the distance to the next hadronic interaction according to the GHEISHA cross-sections. It calls GHESIG and has to be called by GUPHAD. The standard user routines suggested to be able to select the hadronic shower interface at compile time are the following:

```
+DECK,GUPHAD.
      SUBROUTINE GUPHAD
C.
C.    *************************************************************
C.    *                                                           *
C.    *                                                           *
C.    *   GEANT3 user routine called at each step to evaluate     *
C.    *   the remaining distance to the hadronic interaction point*
C.    *                                                           *
C.    *             (TATINA) uses R.Barlow cross-sections tables  *
C.    *             (GHEISHA)uses the GHEISHA cross-sections      *
C.    *                                                           *
C.    *************************************************************
C.
C.
C.     ------------------------------------------------------------
C.
+SELF,IF=-GHEISHA.
      CALL GPHADR
+SELF,IF=GHEISHA.
      CALL GPGHEI
+SELF.
C
      END
+DECK,GUHADR.
      SUBROUTINE GUHADR
C.
C.    **************************************************************
C.    *                                                            *
C.    *     GEANT3 user routine called when a hadronic process     *
C.    *     has been selected in the current step, in order to     *
C.    *     generate the final particle's state                    *
C.    *                                                            *
C.    **************************************************************
C.
C.
C.    -------------------------------------------------------------
C.
+SELF,IF=-GHEISHA.
      CALL GHTATI
+SELF,IF=GHEISHA.
      CALL GHEISH
+SELF.
C
      END
```

The inclusion of these two routines, to be compiled with the PATCHY flag GHEISHA activated, is all the user has to do in order to switch on the GHEISHA code.

CALL GHEISH

This is the main steering routine for the hadronic interactions and is basically a fan out switch yard to the various "cascade" routines of GHEISHA which treat the particular hadronic interaction. Here the kind of interaction that is going to happen is decided via the setting of the INT flag, with the following meaning:

INT	= 0	no interaction (NONE)
	= 1	elastic scattering occurs (ECOH)
	= 2	inelastic incoherent interaction occurs (INHE) 1 and 2 include also nuclear reaction processes at very low energies.
	= 3	nuclear fission with inhelastic scattering occurs (FISS)
	= 4	neutron nuclear captur occurs (CAPT)

Then the corresponding cascade routine is called. Upon exit from this there is a check whether the interaction has generated new particles or not. If yes, the particles generated are copied in the GEANT temporary stack (GKING). If the particle is an heavy fragment or a proton and it is below the energy cut specified via the CUTS data card, it is not stored in the stack but the kinetic energy is collected. There is a limit to 100 particles in the GEANT stack. The user is left to decide in GUSTEP what to do with these new tracks. This routine is called also in case of a stopping particle, i.e. particles with kinetic energy below the GEANT cuts. In this case the routine GHSTOP (see later) is called to handle the stopping particle. The printing flags for GHEISHA are also set in this routine according to the current value of IDEBUG.

This routine should be called by the user routine GUHADR (see above).

The tuning of the final state kinematics as well as the calculation of the cross-sections (see GHESIG above) may be changed by four parameters which are stored sequentially in the bank pointed to by LQ(JTMED−NUMED) (if any). These parameters are empirical corrections which at the moment have the values:

1. 0.7 1.2 0.5

and are currently set in GHEISHA for the following materials:

Material	Component	Rel. number of atoms
BGO	Bi	4
	Ge	3
	O	12
CHK	C	1
	H	1
CH05	C	1
	H	5
CH10	C	1
	H	10
CH50	C	1
	H	50
PBD	C	20
	H	22
	N	2
	O	1
POPOF	C	24
	H	16
	N	2
	O	2
PPO	C	15
	H	11
	N	1
	O	1
NAI	Na	1
	I	1
CSI	Cs	1
	I	1
LII	Li	1
	I	1
CAI2	Ca	1
	I	2
KI	K	1
	I	1
PLEX	C	5
	H	8
	O	2
MYLA	C	5
	H	4
	O	2
AERO	Si	1
	O	4
	H	4

The setting of the parameter for a given tracking medium can be achieved with the following piece of code:

```
CALL GSTPAR(ITMED,'GHCOR1',1.0)
CALL GSTPAR(ITMED,'GHCOR2',0.7)
CALL GSTPAR(ITMED,'GHCOR3',1.2)
CALL GSTPAR(ITMED,'GHCOR4',0.5)
```

The first parameter is actually a flag which, if different from 0 triggers the calculation of the cross-section correction, but in view of future developments it is good practice set it to 1.0 when those corrections are required. ITMED is the tracking medium number as set in GSTMED for which corrections are requested.

CALL GHSTOP

This is an internal routine used to handle stopping particles, called by GHEISH. Here again we have a switchyard to the various routines handling the low energy particles. In particular this routine can lead to nuclear absorption for negative pions and negative kaons (ABSO), to annihilation for antineutrons, and antiprotons (ANNH). The kinetic energy is completely absorbed, and in all the cases not mentioned above (excluding neutrons, protons, deuterons, tritiums and alphas particles) the particle is decayed at rest via the standard GEANT decay routine GDECAY.

Author(s) : F. Bruyant
Origin : R. Brun

Submitted: 15.08.84
Revised: 28.04.86

Introduction to the Tracking package

1. The tracking package

In the context of simulation programs, 'tracking' a particle through matter consists of predicting the spatial coordinates of a set of points which define the trajectory and of computing the components of the momentum at these points. This is usually achieved by integrating the 'equations of motion' over successive ***steps*** and applying corrections when necessary to account for the perturbations introduced by the presence of matter.

The tracking package contains mainly a subprogram which controls, and effectively performs, the tracking for all particles in the current event and for the secondary products which they might eventually generate, plus some tools for storing the space point coordinates computed along the corresponding trajectories.

2. The step size

When tracking particles through a complex medium structure one of the critical tasks is the estimation 'a priori' of the step size. In first approximation this is performed automatically by the program.

For a particle with given energy the step size depends primarily on the intrinsic properties of the particle (mass, charge, lifetime, etc.) and on the characteristics of the current medium. The dependence may be due either to (quasi)continuous processes which usually impose a limit to the interval of integration (energy loss, multiple scattering) or to the occurence of a discrete process which introduces a discontinuity in the trajectory (decay, electromagnetic or hadronic interaction). In addition to these physical effects there are constraints of a geometrical nature, the step being limited by the path length to the medium boundary.

In practice, the step size depends ultimately on a set of tolerances and cuts which should be optimized by the user for the given application, such as:

- the maximum turning angle due to magnetic field permitted in one step,
- the maximum displacement due to multiple scattering in one step,
- the maximum fractional energy loss in one step,
- the accuracy for crossing medium boundaries and
- the minimum step size due to either energy loss or multiple scattering.

These quantities are part of the so called 'tracking medium' parameters. They have to be provided by the user and stored in the data structure JTMED, through the routine GSTMED [CONS]. Usually, this is done together with the initialisation of the geometrical setup. The optimisation is by no means trivial as the economy of computing time should not lead to an unacceptable loss of accuracy.

Other general information required for the computation of the step size is expected to be available in the data structures JPART and JMATE, for the properties of the particles and of the materials, and in the data structure JVOLUM, for the current medium and its geometrical boundaries. The communication between the tracking package and the structure JVOLUM is achieved through the basic subroutines of the geometry package GMEDIA, GNEXT and GINVOL [GEOM].

Some additional information is computed at tracking time such as the probability of occurence of an interaction. For convenience every particle is assigned a 'tracking type', 1 for the gammas, 2 for the electrons and positrons, 3 for the neutral hadrons (and neutrinos!), 4 for the charged hadrons and 5 for the muons. Which physics processes have potentially to be considered for a given particle depends on its tracking type. For the hadrons it depends also, through the subroutine GUPHAD, on which hadronic processes from TATINA, GHEISHA have been selected [PHYS 001].

3. The subroutines GTREVE, GTRACK and GTVOL

At event level the tracking is controlled by the subroutine GTREVE called by the subroutine GUTREV where the user is free to take any other action. GTREVE loops over all tracks present in the structure JKINE and for each track in turn stores its initial parameters and the properties of the corresponding particle in the common block /GCKINE/, then calls the subroutine GUTRAK where the user is free to take any other action than the default which consists of a call to the subroutine GTRACK to track the particle up to the end: stop, decay, interaction or escape. During this phase it may happen that secondary products have been generated and stored by the user, as explained below, in a temporary stack, or in the permanent structure JKINE. Before going to the next track GTREVE performs the tracking of all particles present in the stack.

The subroutine GTRACK loops over all geometrical volumes seen by the current track, first identifying, through the subroutine GMEDIA, the new volume which the particle has reached and storing the corresponding material and tracking medium constants in the common blocks /GCMATE/ and /GCTMED/, then gives control to the subroutine GTVOL which executes the tracking up to the volume boundary (or stops, or looses its identity due to an interaction or a decay).

GTVOL loops over the successive steps inside the current volume. Through the subroutine GNEXT, it computes the geometrical limit for the step, and through either of the tracking type dependent subroutines GTGAMA, GTELEC, GTNEUT, GTHADR, GTMUON and GTNINO, it computes the physical limits for the step and propagates the particle over a distance which does not exceed the smallest of all limits.

4. Magnetic field routines

As mentioned before, the effective propagation of the particles is controlled by the routines GTGAMA, GTELEC, etc., which call GUSWIM. Depending on the value chosen by the user for the tracking medium parameter IFIELD the default routine GUSWIM calls either

- GRKUTA (for inhomogeneous fields, IFIELD = 1), or
- GHELIX (for quasi-homogeneous fields tilted w.r.t. the reference frame, IFIELD = 2), or
- GHELX3 (for one-component fields, IFIELD = 3).

GRKUTA and GHELIX call the default user subroutine GUFLD where the components of the field at the given point are computed. GHELX3 takes the value of the field in the tracking medium parameter FIELDM.

5. Information available at tracking time, and the subroutine GUSTEP

At any moment the current track parameters are available in the common block /GCTRAK/ as well as all variables which have to be preserved by the tracking routines for the control of the step size. In addition a few flags and variables are stored in the common block /GCTRAK/ to record the history of the current step:

The flag INWVOL for instance is initialized to 1 when entering a new volume and set to 0 for all steps inside the volume or to 2 if the particle has reached the volume boundary.
The flag ISTOP is initialized to 0 and set to 1 if the particle looses its identity or to 2 if it stops.
The effect which is responsible for the limitation of the step size as well as the corrective effects which have been applied at the end of the step, if any, are recorded in NMEC words of the mechanism vector LMEC and this is most useful to understand and debug the program.
The total energy loss for the current step is stored in the variable DESTEP.
This information is necessary for the user to take the proper actions in the subroutine ***GUSTEP*** which is called at the end of every step in GTVOL and also when entering a new volume in GTRACK.

In addition, when relevant, the number NGKINE of secondary products which have been generated and their characteristics are stored in the common block /GCKING/ together with an identification (array LMEC) of which process is responsible. Depending on the application and on the particle type, the user may decide in GUSTEP either

- to keep track of the newly produced secondary track. The user then will have to store it in the data structure JKINE where its identity will be preserved, or
- to enter it in a temporary stack, the data structure JSTAK, from which it will be extracted and track-followed without keeping its identity further, or
- to simply forget about it after proper account of its residual energy, at least when the current volume has been declared as a sensitive detector.

6. Connection with the detector response package

The loop over the volumes in GTRACK makes the interface with the detector response package simple [HITS]. By construction of the geometrical setup there is a correspondance between the volumes seen by the particle and the components of the detectors. When entering a new volume (in GTRACK) the subroutine GFINDS is called. If the volume has been declared by the user as a sensitive detector through appropriate calls to GSDET and if the corresponding tracking medium constant ISVOL is non zero, GFINDS returns in the common block /GCSETS/ the information to identify uniquely the detector component. This enables the user in GUSTEP to record the hits in the proper JHITS substructure [HITS].

7. Connection with the drawing package

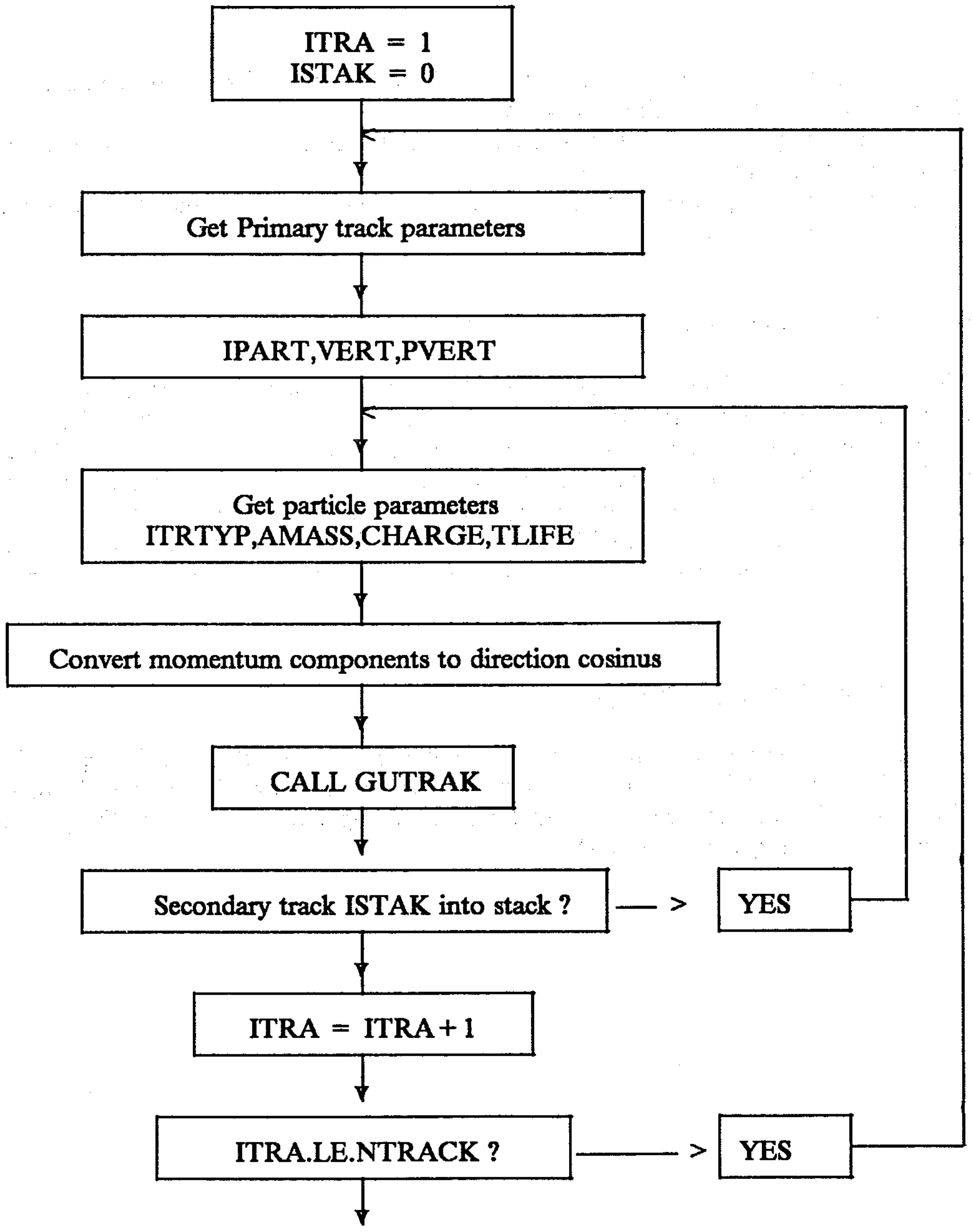
The coordinates of the space points generated during the tracking are available at each step in the common block /GCTRAK/. In GUSTEP the user can store them in the structure JXYZ with the help of the subroutine GSXYZ. This information can be used later for debug (subroutine GPJXYZ) or for the graphical representation of the trajectories [DRAW].

Author(s) : R. Brun
Origin : Same

Submitted: 01.10.84
Revised: 23.05.86

Steering routine to Track one Event

GTREVE Block Diagram


ITRA = 1
ISTAK = 0
Get Primary track parameters
IPART,VERT,PVERT
Get particle parameters
ITRTYP,AMASS,CHARGE,TLIFE
Convert momentum components to direction cosinus
CALL GUTRAK
Secondary track ISTAK into stack ?
YES
ITRA = ITRA+1
ITRA.LE.NTRACK ?
YES

Author(s) : R.Brun
Origin : R.Brun, F.Bruyant, M.Maire

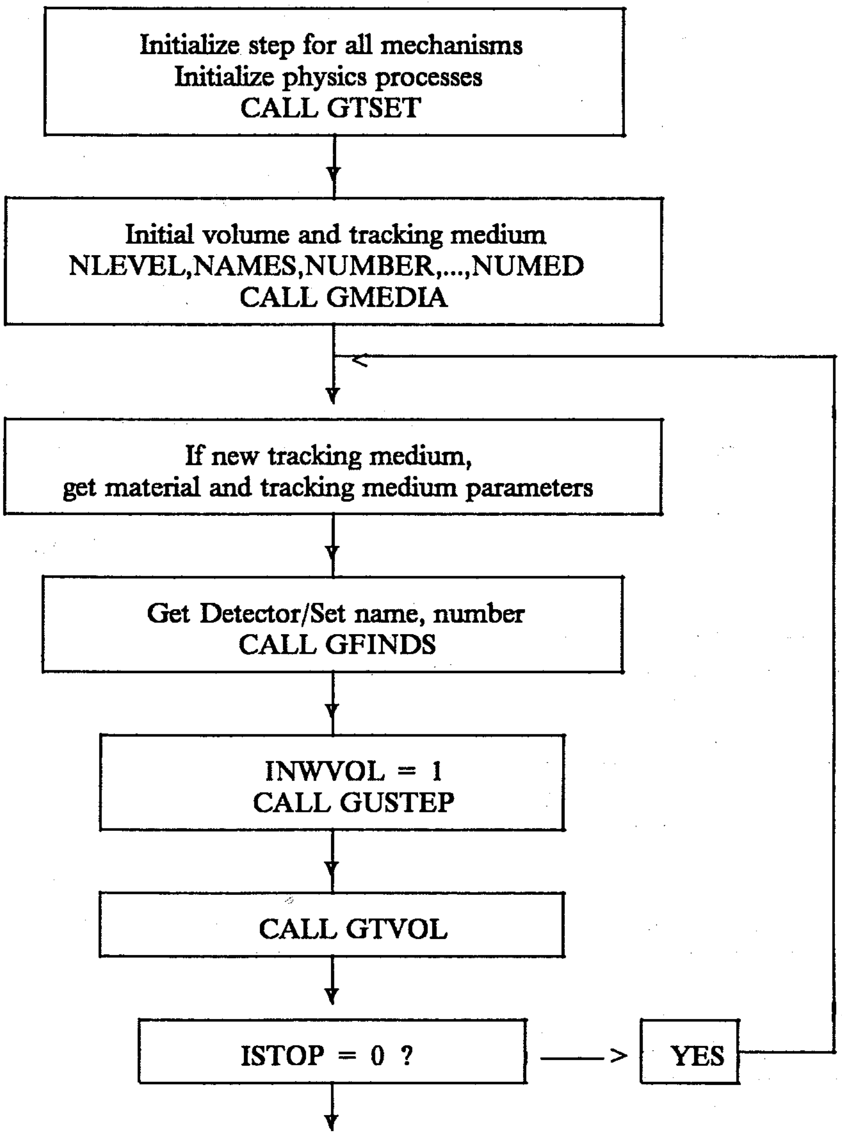
Submitted: 01.10.84
Revised: 26.05.86

Steering routine to Track one Particle

CALL GTRACK Controls the tracking at track level

Called by GUTRAK.

GTRACK Block Diagram

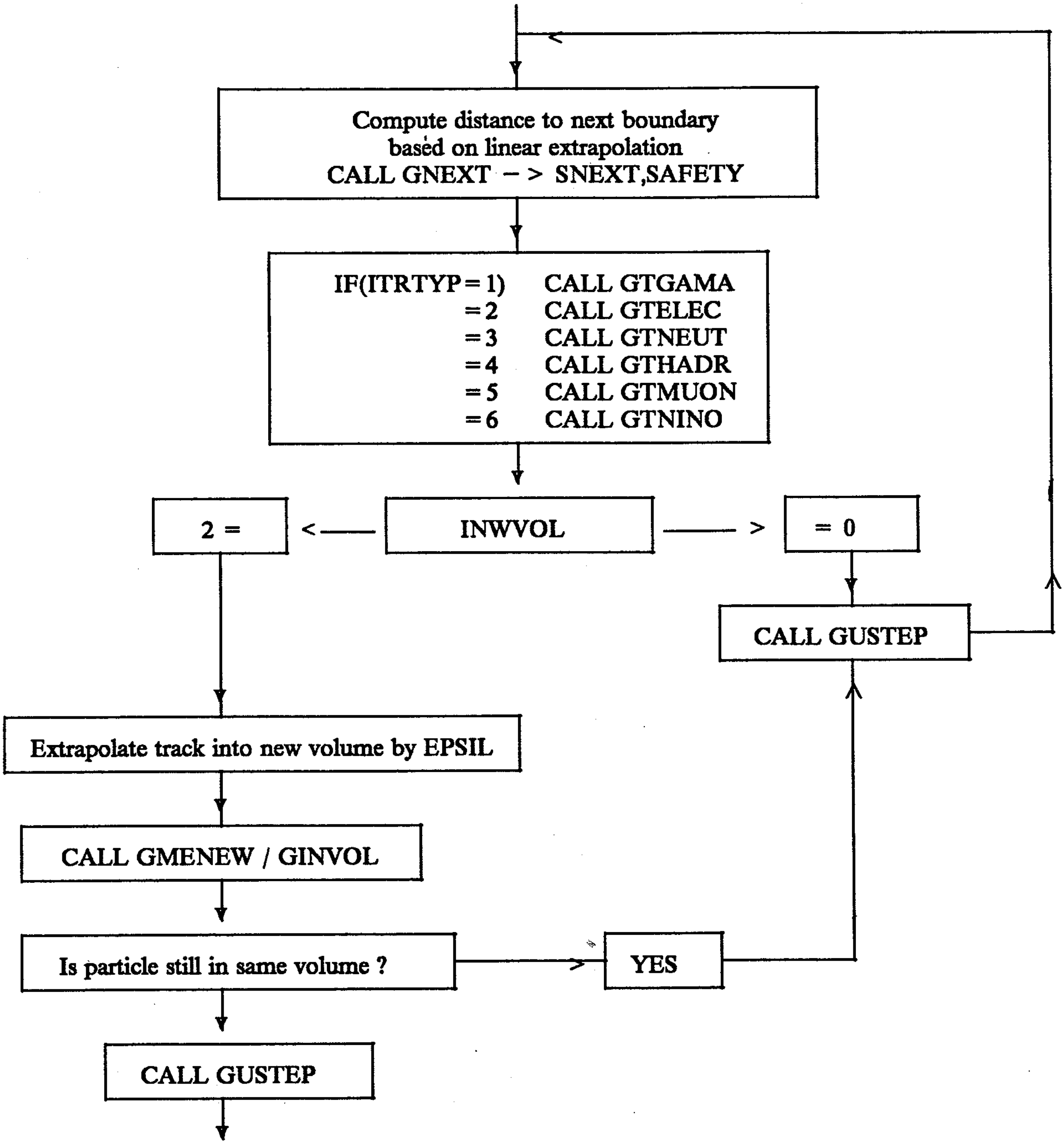

Initialize step for all mechanisms
Initialize physics processes
CALL GTSET
Initial volume and tracking medium
NLEVEL,NAMES,NUMBER,...,NUMED
CALL GMEDIA
If new tracking medium,
get material and tracking medium parameters
Get Detector/Set name, number
CALL GFINDS
INWVOL = 1
CALL GUSTEP
CALL GTVOL
ISTOP = 0 ?
YES

Author(s) : R.Brun
Origin : R.Brun, F.Bruyant, M.Maire

Submitted: 01.10.84
Revised: 26.05.86

Tracking one Particle through a physical Volume

GTVOL Block Diagram


Compute distance to next boundary
based on linear extrapolation
CALL GNEXT -> SNEXT,SAFETY
IF(ITRTYP=1) CALL GTGAMA
=2 CALL GTELEC
=3 CALL GTNEUT
=4 CALL GTHADR
=5 CALL GTMUON
=6 CALL GTNINO
2 =
INWVOL
= 0
CALL GUSTEP
Extrapolate track into new volume by EPSIL
CALL GMENEW / GINVOL
Is particle still in same volume ?
YES
CALL GUSTEP

Author(s) : R.Brun
Origin : Same

Submitted: 01.10.84
Revised: 23.05.86

The Tracking routines

```
CALL GTGAMA
```

```
CALL GTELEC
```

```
CALL GTNEUT
```

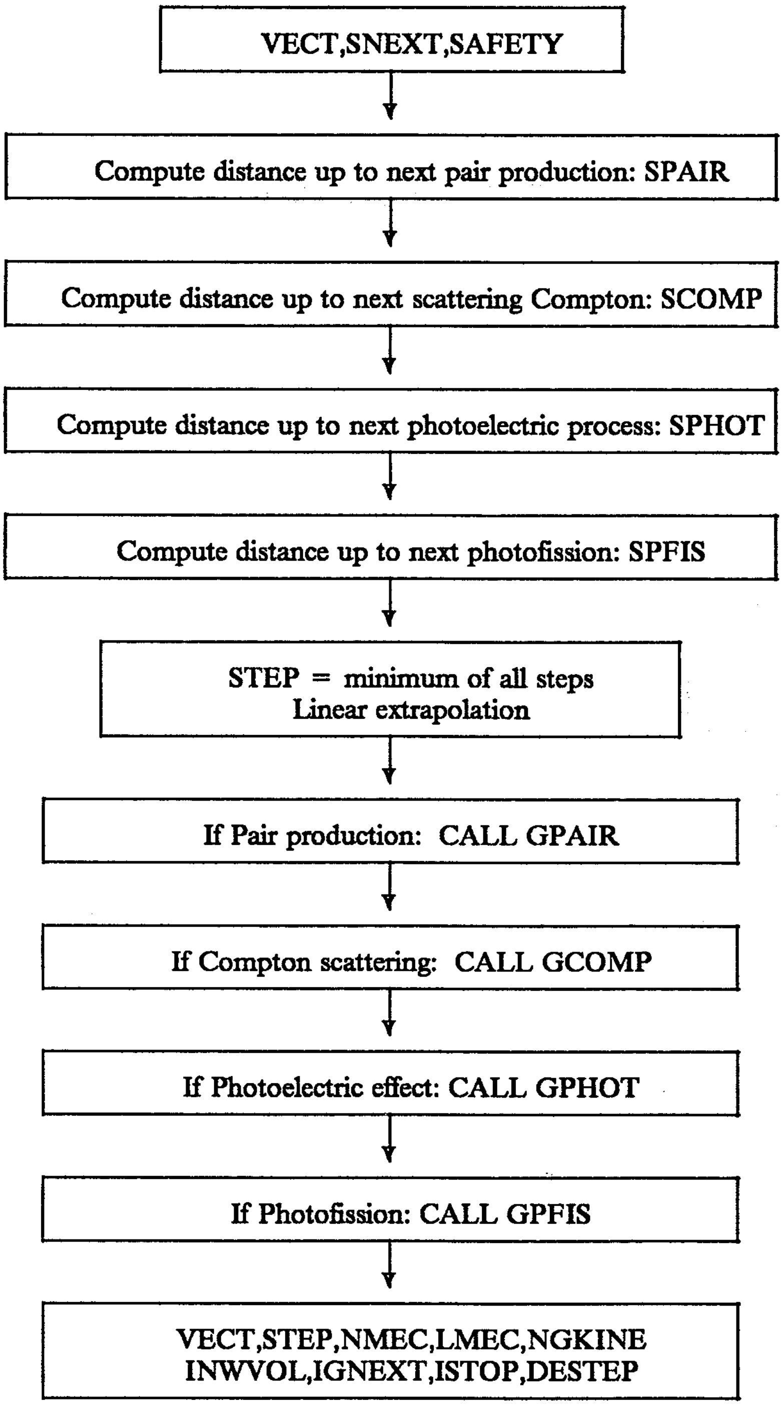
```
CALL GTHADR
```

```
CALL GTMUON
```

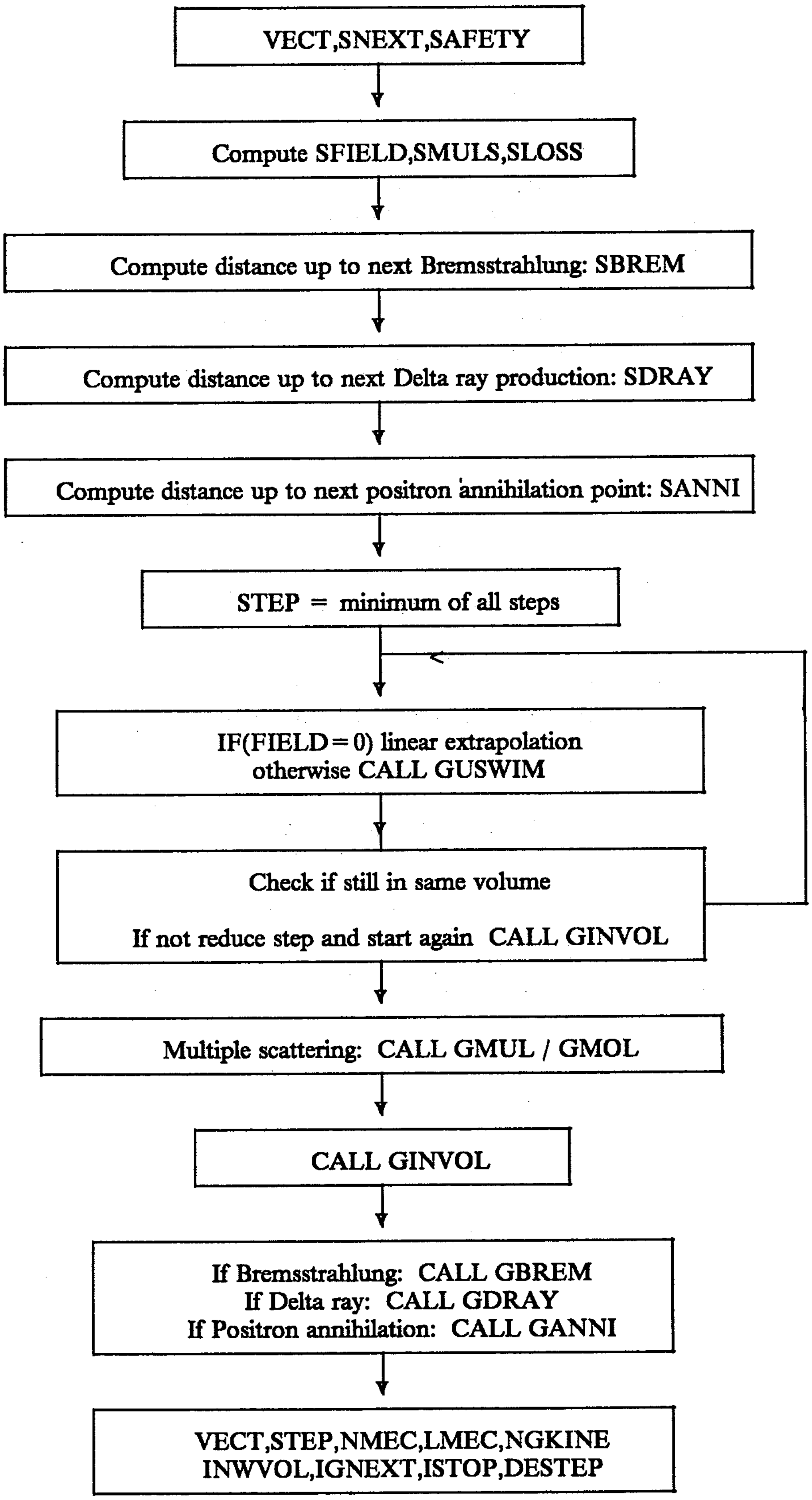
```
CALL GTNINO
```

See block diagrams below.

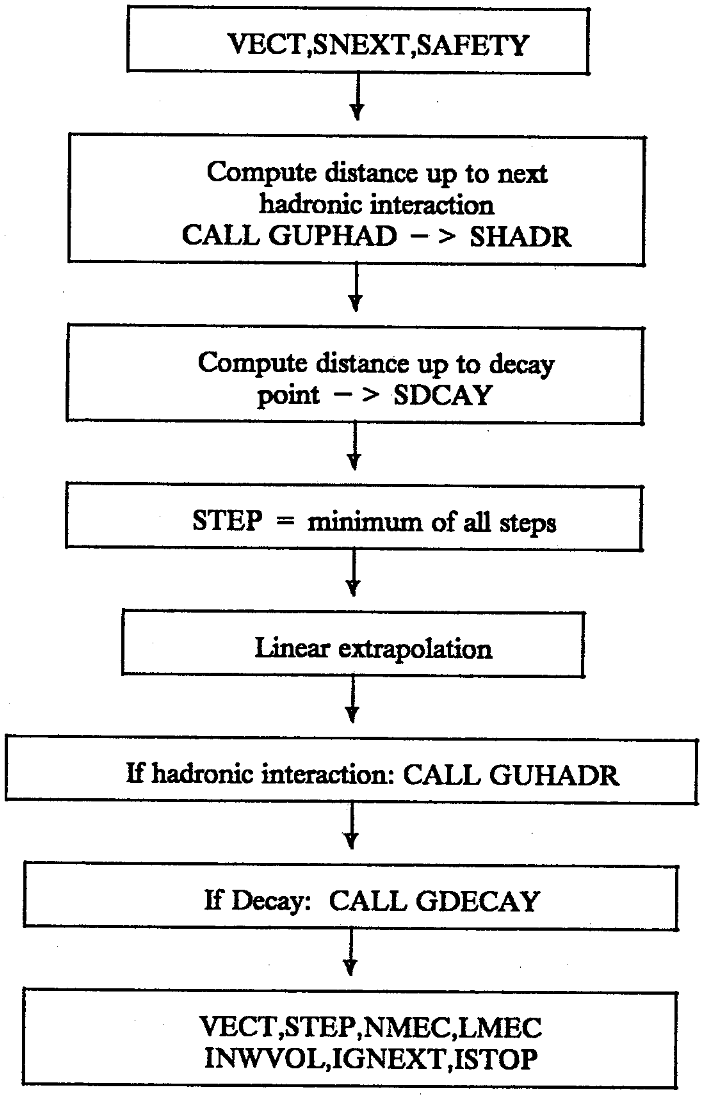
GTGAMA Block Diagram


VECT,SNEXT,SAFETY
Compute distance up to next pair production: SPAIR
Compute distance up to next scattering Compton: SCOMP
Compute distance up to next photoelectric process: SPHOT
Compute distance up to next photofission: SPFIS
STEP = minimum of all steps
Linear extrapolation
If Pair production: CALL GPAIR
If Compton scattering: CALL GCOMP
If Photoelectric effect: CALL GPHOT
If Photofission: CALL GPFIS
VECT,STEP,NMEC,LMEC,NGKINE
INWVOL,IGNEXT,ISTOP,DESTEP

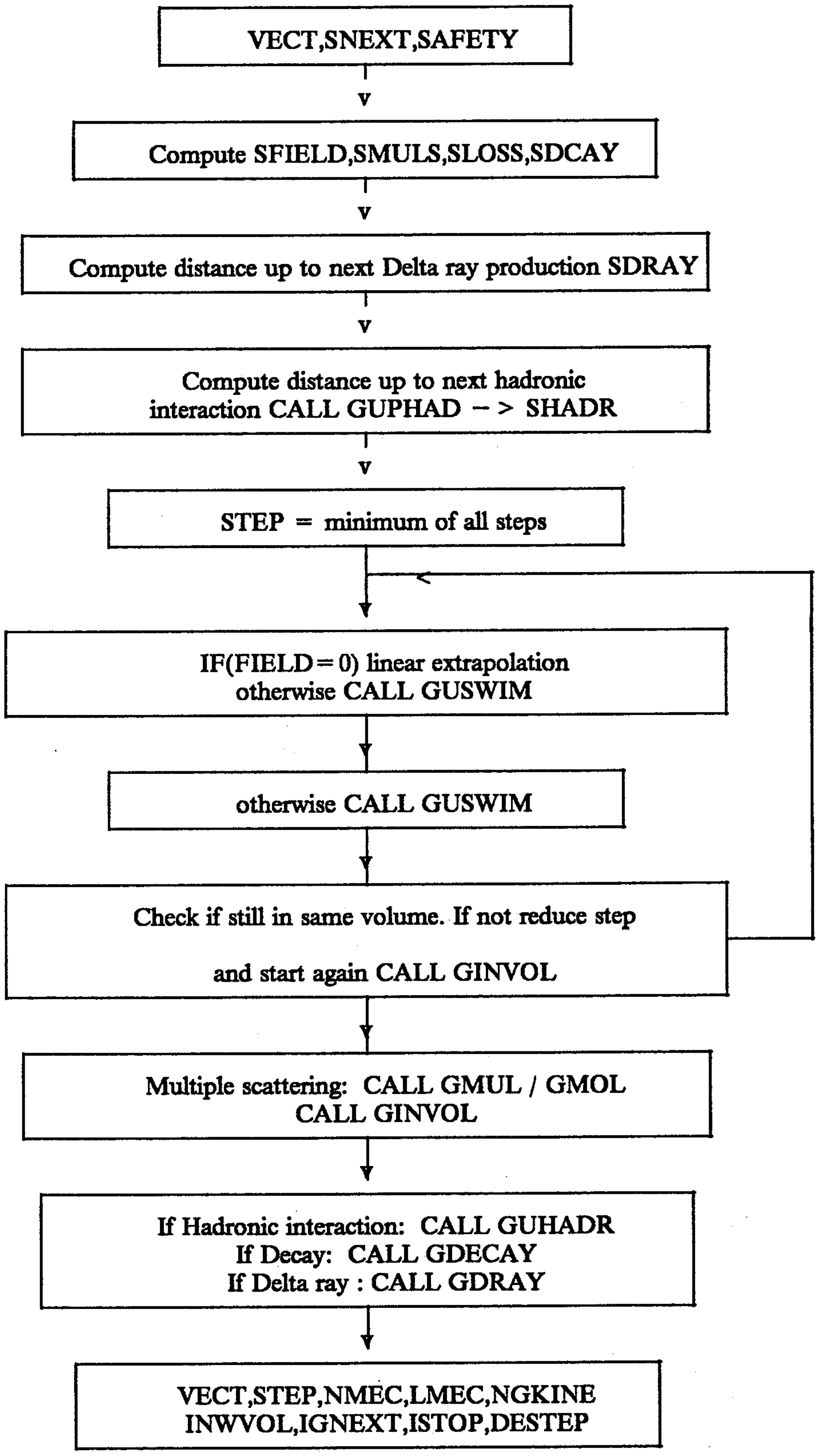
GTELEC Block Diagram


VECT,SNEXT,SAFETY
Compute SFIELD,SMULS,SLOSS
Compute distance up to next Bremsstrahlung: SBREM
Compute distance up to next Delta ray production: SDRAY
Compute distance up to next positron annihilation point: SANNI
STEP = minimum of all steps
IF(FIELD = 0) linear extrapolation
otherwise CALL GUSWIM
Check if still in same volume
If not reduce step and start again CALL GINVOL
Multiple scattering: CALL GMUL / GMOL
CALL GINVOL
If Bremsstrahlung: CALL GBREM
If Delta ray: CALL GDRAY
If Positron annihilation: CALL GANNI
VECT,STEP,NMEC,LMEC,NGKINE
INWVOL,IGNEXT,ISTOP,DESTEP

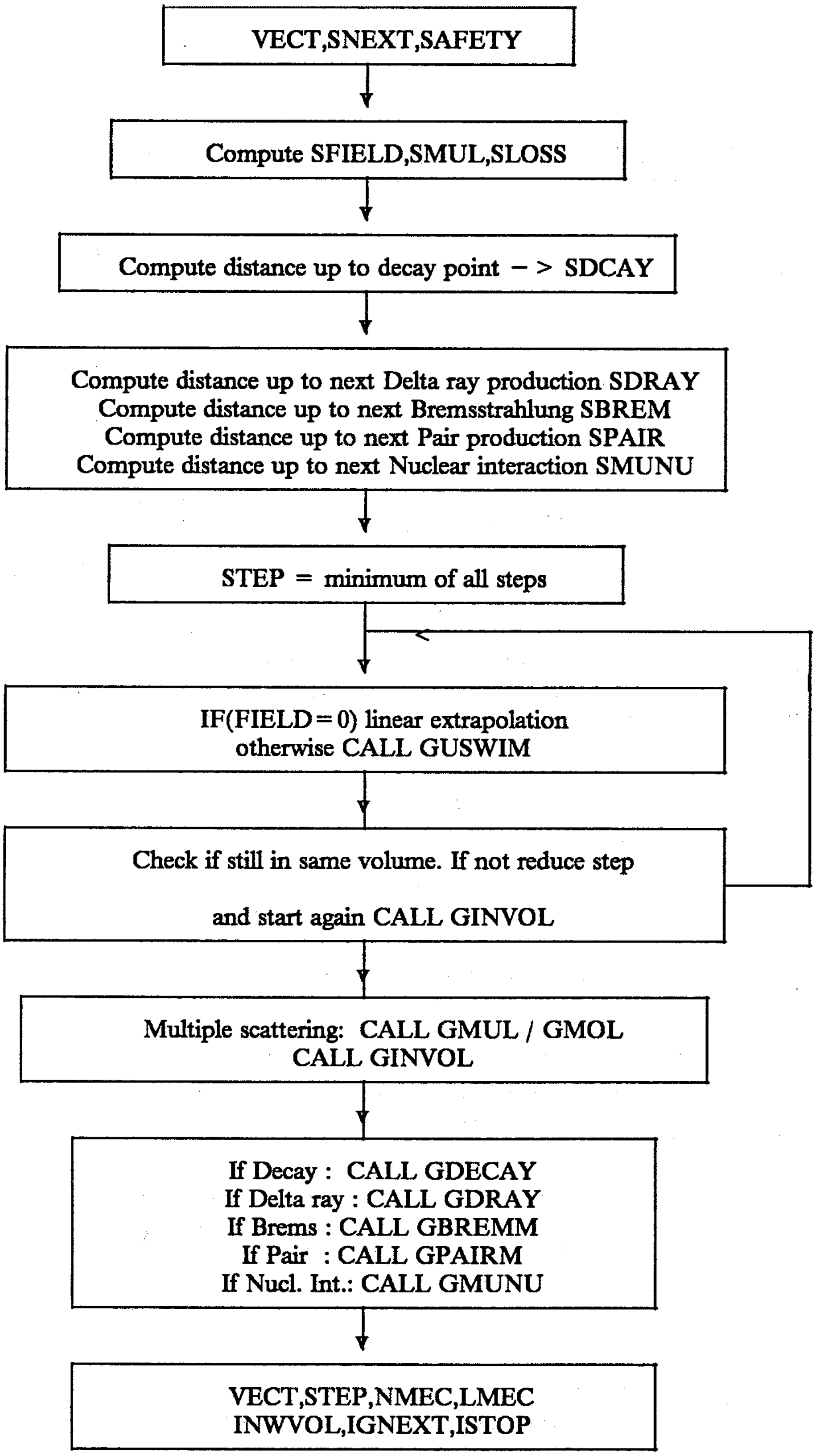
GTNEUT Block Diagram


VECT,SNEXT,SAFETY
Compute distance up to next hadronic interaction CALL GUPHAD − > SHADR
Compute distance up to decay point − > SDCAY
STEP = minimum of all steps
Linear extrapolation
If hadronic interaction: CALL GUHADR
If Decay: CALL GDECAY
VECT,STEP,NMEC,LMEC INWVOL,IGNEXT,ISTOP

GTHADR Block Diagram


VECT,SNEXT,SAFETY
Compute SFIELD,SMULS,SLOSS,SDCAY
Compute distance up to next Delta ray production SDRAY
Compute distance up to next hadronic interaction CALL GUPHAD – > SHADR
STEP = minimum of all steps
IF(FIELD = 0) linear extrapolation otherwise CALL GUSWIM
otherwise CALL GUSWIM
Check if still in same volume. If not reduce step and start again CALL GINVOL
Multiple scattering: CALL GMUL / GMOL CALL GINVOL
If Hadronic interaction: CALL GUHADR
If Decay: CALL GDECAY
If Delta ray : CALL GDRAY
VECT,STEP,NMEC,LMEC,NGKINE
INWVOL,IGNEXT,ISTOP,DESTEP

GTMUON Block Diagram


VECT,SNEXT,SAFETY
Compute SFIELD,SMUL,SLOSS
Compute distance up to decay point − > SDCAY
Compute distance up to next Delta ray production SDRAY
Compute distance up to next Bremsstrahlung SBREM
Compute distance up to next Pair production SPAIR
Compute distance up to next Nuclear interaction SMUNU
STEP = minimum of all steps
IF(FIELD=0) linear extrapolation
otherwise CALL GUSWIM
Check if still in same volume. If not reduce step
and start again CALL GINVOL
Multiple scattering: CALL GMUL / GMOL
CALL GINVOL
If Decay : CALL GDECAY
If Delta ray : CALL GDRAY
If Brems : CALL GBREMM
If Pair : CALL GPAIRM
If Nucl. Int.: CALL GMUNU
VECT,STEP,NMEC,LMEC
INWVOL,IGNEXT,ISTOP

Author(s) : R. Brun
Origin : Same

Submitted: 01.06.83
Revised: 28.04.86

Storing Shower Tracks in the Stack

CALL GSKING(IT)

This routine replaces the old obsolete routine GSSTAK. It is used to save in the stack one or all particles generated during the current step. It can be called for example from GUSTEP.

If IT = 0 then GSKING stores all particles which have been generated. Their number NGKINE and their parameters are obtained from
COMMON /GCKING/ KCASE, NGKINE, GKIN(5,100), TOFD(100)

If IT .NE. 0 then GSKING stores only the track number IT taken from /GCKING/

For each particle saved, the momentum components, as well as the time of flight (TOFG+TOFD(IT)) are stored.

GEANT 3.10 | USER'S GUIDE | TRAK 399

Author(s) : R.Brun
Origin : Same

Submitted: 01.11.83
Revised: 28.05.86

The Shower Stack data structure JSTAK

JSTAK

1	no. of tracks in the stack
2	VX 1st track
3	VY
4	VZ
5	PX
6	PY
7	PZ
8	Energy
9	IPART
10	TOFG
11	VX 2nd track
12	VY 2nd track
13	etc.

The JSTAK bank is filled by the routine GSKING.

Author(s) : R. Brun
Origin : GEANT3

Submitted: 01.06.83
Revised: 28.04.86

Track Space Points

CALL GSXYZ

Stores current space point from COMMON/GCTRAK/ into the data structure JXYZ.

CALL GPJXYZ(NUMB)

NUMB prints space points for track number NUMB
if =0 prints space points for all tracks.

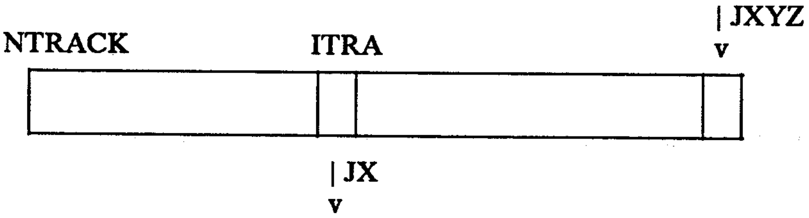
CALL GPCXYZ

Prints the tracking and physics parameters after the current step. This routine can be called from GUSTEP.

Author(s) : R.Brun
Origin : Same

Submitted: 01.11.83
Revised: 28.04.86

The Space Point data structure JXYZ



1	no. of free words in the bank
2	Pointer to current stack track
3	no. of points for the primary track = N
4	IPART Particle code
5	X1, Y1, Z1
8	X2, Y2, Z2
11	etc.
•	no of points for the 1st stack track
•	IPART Particle code
•	X1, Y1, Z1
•	X2, Y2, Z2
•	etc.
•	:

JX = LQ(JXYZ−ITRA) pointer to space points for track number ITRA

The space point banks JXYZ are only used for debug and display purposes. They can be filled by using the routine GSXYZ in the routine GUSTEP for example. The drawing routine GDXYZ gets the space coordinates from JXYZ.

Author(s) : R.Brun, M.Hansroul
Origin : GEANT2/3

Submitted: 01.10.81
Revised: 14.05.86

Tracking routines in a Magnetic Field

CALL GUSWIM(CHARGE,STEP,VECT,VOUT*)

Called by the GEANT tracking routines: GTELEC, GTHADR, GTMUON.

Calls GRKUTA, GHELIX, GHELX3 according to IFIELD = 1,2,3 respectively.
IFIELD = −1 reserved for user decision in GUSWIM.

CHARGE	particle charge
STEP	step size
VECT	initial coordinates, direction cosines, momentum
VOUT	output coordinates, direction cosines, momentum.

CALL GHELX3(FIELD,STEP,VECT,VOUT*)

Performs the tracking (one step) of a particle in a magnetic field. The trajectory is assumed to be a helix in a constant field taken at the mid point of the step. The field is along axis 3.

FIELD	field value multiplied by the charge of the particle
STEP	step size
VECT	initial coordinates, direction cosines, momentum
VOUT	output coordinates, direction cosines, momentum.

The call to this routine is selected by the default routine ***GUSWIM*** when the parameter IFIELD =3 in the tracking medium definition (see GSTMED). When the routine is called by the GEANT tracking routines the parameter FIELD is taken from the FIELDM parameter for this tracking medium (see GSTMED).

CALL GHELIX(CHARGE,STEP,VECT,VOUT*)

Performs the tracking (one step) in a magnetic field. The trajectory is assumed to be a helix in a constant field taken at the mid point of the step.

CHARGE	particle charge
STEP	step size
VECT	initial coordinates, direction cosines, momentum
VOUT	output coordinates, direction cosines, momentum.

The call to this routine is selected by the default routine *GUSWIM* when the parameter IFIELD = 2 in the tracking medium definition (see GSTMED). The following user routine is called

```
CALL GUFLD(VECT,F*)
```

where, F is a vector containing the Field components.

```
CALL GRKUTA(CHARGE,STEP,VECT,VOUT*)
```

Runge−Kutta method for tracking a particle through a magnetic field. Uses Nystroem algorithm (see Handbook NAT.BUR.OF standards, Procedure 25.5.20)

CHARGE	particle charge
STEP	step size
VECT	initial coordinates, direction cosines, momentum
VOUT	output coordinates, direction cosines, momentum.

The call to this routine is selected by the default routine *GUSWIM* when the parameter IFIELD = 1 in the tracking medium definition (see GSTMED).

For one step 3 or 4 calls to a user provided routine *GUFLD* are done.

```
CALL GUFLD(VECT,F*)
```

where, F is a vector with elements containing the field components.

Author(s) : R. Brun
Origin : Same

Submitted: 15.06.84
Revised: 30.05.86

Introduction to the Interactive version of GEANT3

The interactive version is an essential tool for users who are in charge of the design of a detector in a Physics collaboration. In addition to all the BATCH tools which are also available, the interactive user can call one by one and in any order all the basic functions of GEANT to :

- Design or modify the geometry of the setup
- Exploit the drawing package in a more efficient way.
- Change the running conditions on a event by event basis.

The system is based on the ZCEDEX[1] command processor and is organized into 4 menus:

MENU 1	Drawing commands
MENU 2	Graphics control commands
MENU 3	Geometry commands
MENU 4	General control commands

The ZCEDEX BREAK capabilities may be used, and the interactive routine GINTER is designed in such a way that it can be called from the symbolic debugger. The user has to know a minimum about ZCEDEX and should read at least the first chapter of the manual. The following write-ups describe individual commands which can be typed one by one at the terminal, or grouped into macros which can be edited and saved in the ZCEDEX environment. Optional parameters are shown within brackets [param].

This interactive version has so far been implemented and tested under

VAX/VMS , IBM/VM-CMS and APOLLO/AEGIS

[1] ZCEDEX User Guide – CERN DD/EE/80-6

Author(s) : R.Brun, P.Zanarini
Origin : Same

Submitted: 15.06.84
Revised: 22.04.86

The Drawing Commands

```
DRAW  name [theta phi psi u0 v0 su sv]
```

CALL GDRAW(name,theta,phi,psi,u0,v0,su,sv)

If optional parameters are missing, the current values in /GCDRAW/ are taken

```
DVOLUME  n lnames lnumbs nrs [theta phi psi u0 v0 su sv]
```

CALL GDRVOL(n,lnames,lnumbs,nrs,theta,phi,psi,u0,v0,su,sv)

N is number of levels from the top of the geometry structure until the volume lnames(n),lnumbs(n) to be drawn

LNAMES and LNUMBS are arrays containing the names and numbers identifying the path

NRS is the reference system used: NRS = 0 for MARS or NRS < > 0 for DRS

If optional parameters are missing, the current values in /GCDRAW/ are taken

```
DCUT name iaxis cutval [u0 v0 su sv]
```

CALL GDRAWC(name,iaxis,cutval,u0,v0,su,sv)

the cut-plane is normal to iaxis (1,2,3) and placed at the distance cutval from the origin. The resulting picture is seen from the the same axis.
If optional parameters are missing, the current values in /GCDRAW/ are taken

```
DXCUT name cutthe cutphi cutval [theta phi u0 v0 su sv]
```

CALL GDRAWX(name,cutthe,cutphi,cutval,theta,phi,u0,v0,su,sv)

the cut-plane is normal to the line given by the cut angles cutthe,cutphi and placed at the distance cutval from the origin. The resulting picture is seen from the viewing angles theta,phi.

If optional parameters are missing, the current values in /GCDRAW/ are taken

DTREE [name levmax isel]

CALL GDTREE(name,levmax,isel)

isel is used to select options in the picture of the tree:

if 0, then we draw only node name

if xxxxx1, then we add multiplicity

if xxxx1x, then we add 'ONLY' information

if xxx1xx, then we add 'DET ' information

if xx1xxx, then we add 'SEEN' information

if x1xxxx, then we add a little picture of the volume, above each node

if 1xxxxx, then we add the graphics cursor picking capability :

after the cursor has been moved press ...

' ' (space bar) to redraw the tree from the pointed node

'2' to set the SEEN attribute -2

'−' to set the SEEN attribute −1

'0' to set the SEEN attribute 0

'1' to set the SEEN attribute 1

Defaults are : name= <global detector name> levmax=0 isel=111

DSPEC name

CALL GDSPEC(name)

DFSPC name [isort inter]

CALL GDFSPC(name,isort,inter)

If isort=1 all the pictures will be drawn in ascending alphabetic order.

If inter=1 the routine will prompt the user at each plot before doing a clear screen, otherwise it will clear automatically the screen before starting a new frame.

Defaults are : isort=0 inter=1

DTEXT x0 y0 text size angle lwidth iopt

CALL GDRAWT(x0,y0,text,size,angle,lwidth,iopt)

DVECTOR x-vect y-vect npoint

CALL GDRAWV(x-vect,y-vect,npoint)

where x-vect and y-vect are 2 ZCEDEX vectors

DSCALE u v

CALL GDSCAL(u,v)

DAXIS x0 y0 z0 dx

CALL GDAXIS(x0,y0,z0,dx)

DMAN u v

CALL GDMAN(u,v)

DHEAD [isel name chrsiz]

CALL GDHEAD(isel,name,chrsiz)

isel is an option to be selected for the title name :

isel = 0 to have only the header lines
isel = xxxxx1 to add the text name centered on top of header
isel = xxxx1x to add global detector name (first volume) on left
isel = xxx1xx to add date on right
isel = xx1xxx to select thick characters for text in top of header
isel = x1xxxx to add the text 'EVENT NR x' on top of header
isel = 1xxxxx to add the text 'RUN NR x' on top of header

Note that isel = x1xxx1 or isel = 1xxxx1 are illegal choices, i.e. they generate overwritten text.

name is the title (string terminated by a dollar sign) and chrsiz the character size in cm of text name

Defaults are : isel = 111110 name = '$' chrsiz = 0.6

MEASURE

Position the cursor on first point (u1,v1) and press the space bar.

Position the cursor on second point (u2,v2) and press the space bar.

The command will compute the distance in space separating the 2 points on the projection view.

ZOOM [zfu zfv isel uz0 vz0 u0 v0]

CALL GDZOOM(zfu,zfv,uz0,vz0,u0,v0)

This command sets the zoom parameters that will be used by next calls to the drawing routines. Each zoom operation is always relative to the status of the current zoom parameters. The scale factors in u,v are respectively zfu,zfv. zfu=0 (or zfv=0) will act as a reset (i.e. unzoomed viewing). The zoom is computed around uz0,vz0 (user coordinates), and the resulting picture will be centered at u0,v0.

If isel=0 : 1. position the cursor at (uz0,vz0)
2. press the space bar

(u0,v0 are chosen at centre of screen)

If isel=1 : 1. position the cursor at first corner of zoom rectangle
2. press the space bar
3. position the cursor at second corner of zoom rectangle
4. press the space bar

(zfu,zfv are chosen according to zoom rectangle;
uz0,vz0 are chosen at centre of zoom rectangle;
u0,v0 are chosen at centre of screen)

If isel=2 : 1. position the cursor at (uz0,vz0)
2. press the space bar
3. position the cursor at (u0,v0)
4. press the space bar

Defaults are : zfu=2. zfv=zfu
uz0,vz0= <centre of screen>
u0,v0= <centre of screen>

DXYZ [itra]

CALL GDXYZ(itra)

KXYZ [epsilo]

CALL GKXYZ(epsilo)

Defaults are : epsilo = 0.25

The picking of track points requires the JXYZ data structure and is repeated until the character typed is 'Q' or 'q'.

EPSILO is the delta angle used for pick; if EPSILO = 0 there is no optimization performed and over all the track points the one nearest to the pick point is taken.

DPART [itra isel size]

CALL GDPART(itra,isel,size)

isel = x1 to draw the track number
isel = 1x to draw the particle name

Defaults are : itra = 0 isel = 11 size = 0.25

DHITS [iuset iudet itra isymb ssymb]

CALL GDHITS(iuset,iudet,itra,isymb,ssymb)

The character plotted at each hit-point may be chosen by isymb :

−1	(small) hardware points	(fast)
0	software crosses	(default)
840,850	empty/full circles	(slow)
841,851	empty/full squares	(slow)
842,852	empty/full triangles (up)	(slow)
843,853	empty diamond/full triangle (down)	(slow)
844,854	empty/full stars	(slow)

Except for isymb = −1 the size of the character on the screen can be chosen by ssymb cm (default = 0.1).

KHITS [iuset iudet epsilo]

CALL GKHITS(iuset,iudet,epsilo)

Defaults are : iuset = 0 iudet = 0 epsilo = 0.1

The picking of hit points requires the appropriate JSET data structure and is repeated until the character typed is 'Q' or 'q'. If the character typed to pick is 'K' or 'k' then the kinematics of the corresponding track is also printed.

The search is done over all the hits of all tracks in detector IUDET of set IUSET.

EPSILO is the pick aperture; if EPSILO < 0 its absolute value is taken and in addition the pick aperture is drawn; if EPSILO = 0 there is an infinite pick aperture and over all the hits the one nearest to the pick point is taken.

```
DCHIT  [iuset iudet itra isymb sizmax ihit hitmin hitmax]
```

```
CALL GDCHIT(iuset,iudet,itra,isymb,sizmax,ihit,hitmin,hitmax)
```

The character plotted at each hit-point may be chosen by isymb :

−1	(small) hardware points	(fast)
0	software crosses	(default)
840,850	empty/full circles	(slow)
841,851	empty/full squares	(slow)
842,852	empty/full triangles (up)	(slow)
843,853	empty diamond/full triangle (down)	(slow)
844,854	empty/full stars	(slow)

Except for isymb = −1 the SIZE of the character on the screen is a function of HITS(IHIT), with HITMIN and HITMAX defining the range. The maximum character size (used in overflow) is SIZMAX.

SIZE = sizmax * (hits(ihit) − hitmin) / hitmax

Defaults are : iuset = 0 iudet = 0 itra = 0 isymb = 0
sizmax = 1 ihit = 4 hitmin = 0 hitmax = 0

```
DUVIEW  name type cpxtyp [iview]
```

```
CALL GUVIEW(name,type,cpxtyp,iview)
```

Author(s) : R. Brun, P.Zanarini
Origin : Same

Submitted: 15.06.84
Revised: 22.04.86

The Graphic Control Commands

```
CAPTURE_FILE lun
```

Set ON/OFF the graphics file (PIGS capture file or GKS metafile)
CAPT 10 Output on screen and unit 10 (open graphics file)
CAPT 0 Output on screen only (close graphics file)
CAPT −10 Output on unit 10 only (open graphics file)

lun = 11,12,13 are reserved by ZCEDEX and cannot be used

```
DOPEN iview
```

CALL GDOPEN(iview)

```
DSHOW [iview]
```

CALL GDSHOW(iview)

Default : iview = <iview of last DSHOW command>

```
DELETE iview
```

CALL GDELET(iview)

```
DCLOSE
```

CALL GDCLOS

SATT name iopt val

CALL GSATT(name,iopt,ival)

name = 0 stands for all the volumes

iopt can be chosen among the following :

'WORK' 0 = volume name is inactive for the tracking step
1 = volume name is active for the tracking step (default)
'SEEN' 0 = volume name is set invisible
1 = volume name is set visible (default)
− 1 = name invisible with all its descendants in the tree
− 2 = name visible but not all its descendants in the tree
− 3 = used to specify a special optimization by GDTREE
'LSTY' line style 1,2,3,... (default = 1)
'LWID' line width 1,2,3,... (default = 1)
'COLO' colour code 1,2,3,... (default = 1)
'FILL' fill area 0,1,2,... (default = 0)
'SET ' set number associated to volume name
'DET ' detector number associated to volume name
'DTYP' detector type (1,2)
'NODE' < > 0 = a node is created for 3D (PIONS,etc.)

SCALE gscu gscv

Change the scale factors GSCU and GSCV in /GCDRAW/

COLOR icol

CALL GDCOL(− abs(icol))

LWID lwidth

CALL GDLW(− abs(lwidth))

NEXT

Clear screen (start a new picture on graphics file, if opened)

DOPT [iopt ival]

CALL GDOPT(iopt,ival)

Type DOPT without parameters to get the list of all currently available options.
Type DOPT IOPT to enquire on the current setting of that option.

Defaults are : iopt = 0 ival = 0

RANGE – DISPLAY [x1 y1 x2 y2]

CALL TVRNG('DISPLAY',x1,y1,x2,y2)

Defaults are : x1 = 130. y1 = 10. x2 = 890. y2 = 770.

RANGE – USER [x y]

CALL TVHRNG(x,y)

Defaults are : x = 20. y = x

PIONS – OPEN

Copy all the subsequent drawings on the PIONS structure

PIONS – CLOSE

No more drawings will be added to the PIONS structure

PIONS – GO

Give the control to PIONS, i.e. to its graphics menu

GMR – OPEN

Copy all the subsequent drawings on the GMR structure

GMR – CLOSE

No more drawings will be added to the GMR structure
and the control is given to GMR, i.e. to its graphics menu

SPERS dpers

Set the variable dpers in /GCDRAW/, representing
the distance from the origin when using option PERSpective

MAP – COLOR [icadd icval]

Sets the color table LOOKTB(ICADD) = ICVAL
If ICADD = 0 then LOOKTB(1:16) is taken
If ICVAL is omitted the current value of LOOKTB(ICADD) is shown

Defaults are : icadd = 0 (looktb(i) = i, i = 1,16)

Author(s) : R. Brun, P.Zanarini
Origin : Same

Submitted: 15.06.84
Revised: 14.05.86

The Geometry Commands

***SVOLU** name shape numed par npar*

CALL GSVOLU(name,shape,numed,par,npar,ivolu)

where par is a ZCEDEX vector

***SPOS** name number mother x0 y0 z0 irot only*

CALL GSPOS(name,number,mother,x0,y0,z0,irot,only)

***SDVN** name mother ndiv iaxis*

CALL GSDVN(name,mother,ndiv,iaxis)

***SROTM** irot theta1 phi1 theta2 phi2 theta3 phi3*

CALL GSROTM(irot,theta1,phi1,theta2,phi2,theta3,phi3)

***STMED** ntmed name nmat isvol ifield fieldm tmaxfd tmaxms epsil*

```
 CALL GSTMED(ntmed,name,nmat,isvol,ifield,fieldm,tmaxfd,
+            dmaxms,deemax,epsil,stmin,0,0)

 Defaults are : ntmed=1  nmat=1  isvol=0  ifield=0  fieldm=0.
       tmaxfd=0.01  dmaxms=0.01  deemax=0.01  epsil=0.01
```

EDITV [isel]

CALL GEDITV(isel)

If ISEL is not given, the menu is shown.

When the routine asks for input parameters that have not to be changed, the carriage return must be typed.

ISEL is used to select the editing operation to be performed:

ISEL = 1, to modify shape parameters PAR given by GSVOLU

ISEL = 2, to modify NAME given by GSVOLU

ISEL = 3, to delete NAME given by GSVOLU

ISEL = 4, to unlink NAME,NR given by GSPOS/GSDVN/GSDV..

ISEL = 5, to modify X0,Y0,Z0 of NAME,NR given by GSPOS

ISEL = 6, to modify IROT of NAME,NR given by GSPOS

ISEL = 7, to modify NDIV given by GSDVN

ISEL = 8, to modify IAXIS given by GSDVN

Author(s) : R. Brun
Origin : Same

Submitted: 15.06.84
Revised: 09.09.87

The General Control Commands

***KINE* [*ikine pkine*]**

Set the variables in /GCFLAG/ IKINE, PKINE(10)

Default : ikine = 1

TRACKING

Restart tracking, clearing the tracking and hits banks, but keeping the kinematics.

TRIGGER

Start a new event

SWITCH iswitch ival

Change one element of array ISWIT(10) in /GCFLAG/

***DZSURV bank_name* [*bank_number*]** See ZEBRA User Guide

CALL DZSURV('DZSURV',ixstor,lzloc(ixstor,bank_name,bank_number))

***DZSHOW bank_name* [*bank_number chopt*]** See ZEBRA User Guide

CALL DZSHOW('DZBANK',ixstor,lzloc(ixstor,bank_name,bank_number),chopt ,0,0,0,0)

Defaults are : bank_number = 0 chopt = 'BHV'

DZSNAP [***idiv chopt***] See ZEBRA User Guide

```
      CALL DZSNAP('DZSNAP',idiv,chopt)
```

Defaults are : idiv = 2 chopt = 'M'

DZVERI [***idiv chopt***] See ZEBRA User Guide

```
      CALL DZVERI('DZVERI',idiv,chopt)
```

Defaults are : idiv = ixstor chopt = 'CLSU'

DZAREA [***chopt ixstor***] See ZEBRA User Guide

```
      CALL DZ('DZAREA',ixstor,chopt,lla,'N')
```

Defaults are : ixstor = 0 chopt = ' '

DZSTOR [***ixstor***] See ZEBRA User Guide

```
      CALL DZSTOR('DZSTOR',ixstor)
```

Defaults is : ixstor = 0

MZLOGL level See ZEBRA User Guide

```
      CALL MZLOGL(0,level)
```

Default is : level = 0

FZLOGL lun level See ZEBRA User Guide

```
      CALL FZLOGL(lun,level)
```

Default is : lun = 2 level = 0

***PRINT name* [*number*]**

CALL GPRINT(name,number)

Default number = 0

UNIT lout

to change lout in /GCUNIT/
Note: 5 and 11 are reserved units

***PHITS* [*iuset iudet*]**

to call GPHITS(iuset,iudet)
Defaults are : iuset = 0 iudet = 0

***PDIGI* [*iuset iudet*]**

to call GPDIGI(iuset,iudet)
Defaults are : iuset = 0 iudet = 0

PRMAT imate ipart mecan

CALL GPRMAT(imate,ipart,mecan,90,elow)

***PLMAT imate ipart mecan* [*idm*]**

CALL GPLMAT(imate,ipart,mecan,90,elow,idm)

Default is : idm = 0

PSTAT

CALL GPSTAT

RETURN

Return from the interactive routine GINTER.

Useful when called from the symbolic debugger.

DEBUG [*ideb*]

If ideb = 'ON ' then : idebug = 1, idemin = 1, idemax = 1000000

else : idebug = 0, idemin = 0, idemax = 0

Default ideb = 'ON '

Author(s) : F.Carminati
Origin :

Submitted: 01.10.84
Revised: 06.09.87

Current status of GEANT3

See GEANT NEWS on CERNVM, or on VXCRNA.

Author(s) : R.Brun, F.Bruyant
Origin : J-L.Dekeyser, E.J.Luit

Submitted: 01.10.84
Revised: 30.05.86

Adaptation of GEANT3 to 3081/E Emulators

Starting fall 1985 some work has been done at CERN in view of making use of 3081/E emulators for massive parallel GEANT3 production.

J-L. Dekeyser (CERN/DD/EE) and E.J. Luit (CERN/EP/L3) have been both working full time on this project with, on the one hand, the possibility to exchange information regularly and, on the other hand, the freedom to develop and propose approaches which differ here or there according to the assumptions and criteria that they believe to be more realistic or more efficient.

Their work is described in the following documents to be published shortly, then available on request to their authors :

- GEANT3 parallel production on 3081/E emulators, J-L. Dekeyser, CERN/DD/EE
- Implementation of SIGEL3 on the 3081/E Emulators, E.J. Luit, CERN/EP/L3-405.

A summary of the respective approaches is included below. There is some hope that the experience one will acquire in the coming months will permit a unified treatment for what concerns the modifications to some 'high level' package subroutines (ZEBRA, I/O) and eventually to some GEANT3 subroutines.

Before describing the details of the implementations, a clarifying comment is needed. Since the emulators cannot (at present) do direct I/O to or from disk or tape, I/O is accomplished using buffers on the host and on the emulators. The centralisation of the GEANT3 line printer output, through the subroutine GMAIL, and of the event I/O, through thc subroutincs GGET and GSAVE (both calling FZ-ZEBRA routines), limits the number of places where changes have to be introduced in the program and makes the job somewhat simpler. The modifications introduced in the FZ-routines will have to be reconsidered when the exchange-mode will be operational.

The active cooperation of all members of the Emulators Project group, under the leadership of A. Fucci, and their continuous interest in developing a more user friendly environment, have been essential during this period and should be particularly acknowledged.

The GEANX pam file contains a simple example of application, produced by J-L. Dekeyser. It is proposed as a guide line for people interested in the matter. The code developed by E.J. Luit will be made available at the next release of the L3-software pam files, early Summer 1986.

Appendix 1 (by J-L. Dekeyser)

In the framework of a project to study the various possibilities of GEANT3 parallel production we have chosen to implement a distributed solution which consists of simulating packs of events on independent computers and of restructuring the data after processing. The hardware used for this implementation is a set of 3081/E emulators controlled by an IBM 4361 host computer.

The proposed algorithm is simple. It consists of performing the initialisation phase of GEANT3 on the host computer, then to send the results to each emulator and to start on each emulator the execution of GRUN for a bunch of N/n events, where N is the total number of events and n the number of active emulators. The histograms are reassembled on the host when all emulators are finished.

The transfer of data from the 4361 to the 3081/E and from the 3081/E to the 4361 is made through two separate buffers.

The size of the output buffer can be adjusted according to the resources and may contain either part of an event or one or several events. In case the buffer is not large enough for a complete event a reservation of the buffer for the current emulator is made in GSAVE so as to prevent mixing of data from different emulators.

More details can be found in the in-line comments of the test program proposed in the Patch GEMUL (Pam GEANX310).

Appendix 2 (by E.J. Luit)

Summary of approach chosen for the L3 simulation program SIGEL3

The host uses one buffer for input to the emulators and one buffer for output from the emulators. Only the output buffer will be a dedicated I/O buffer. On the emulator, only one buffer is used. The SYE package, which handles the communication with the emulators, allows one to transfer data from any address on the host to any address on the emulators. This enables the use of two separate buffers on the host and only a single one on the emulators.

The initialisation of the detector is done on the host computer which then writes out the initialisation structures in the output buffer. This is the only instance in which the host writes into the output buffer. The buffer is broadcasted to all selected emulators together with the common blocks that were initialized by BLOCK DATA and the emulators are started. The emulators initialize by reading the initialisation file from the buffer using GGET.

When the initialisation is finished, the emulators halt when entering the event loop in GRUN. This is the only place where program execution on the emulators is interrupted. Meanwhile the host has read in an event from tape or disk, when pregenerated events are used or in case of partial event reprocessing. Otherwise, the host does nothing. The event input is stored in the input buffer.The event is sent to the 3081/E that has halted and the emulator resumes execution. The host prepares the next event and services the next emulator. This goes on till each emulator has been serviced and is processing its first event. The emulators are writing the output event data structures in the buffer.

Now, the main event loop starts. The host (when relevant) prepares an event in the input buffer and waits for any emulator to halt. The part of the buffer on the emulator that is filled is transported to the host output buffer the event input is transported to the emulator that halted and the emulator resumes execution. The host then writes out the buffer on disk or on tape, prepares the next event and waits for the next emulator to halt. The host also takes care of the event numbering. Notice that all I/O that the host performs is done before and after the emulator is serviced. In this way, effective use is made of the computing power of the joint system.

When the required number of events has been sent to the emulators, the host merely gets back the data from the emulators that are still active and writes the last events out till all emulators are finished. After this, the histograms are recuperated by adding the contents of the histograms made by the different emulators.

Two improvements to the approach described above are being investigated. The aim is first to reduce the size of the host job to a bare minimum, then to reduce the size of the buffer on the emulators without deterioration of the speed of the joint system.

The main changes that are required are in the following routines:

MAIN : The common blocks that are initialized by block data, the common GCFLAG and the common that contains the (dedicated) I/O buffer are declared here to get them in the same sequence on the host and the 3081/E so that they can be transferred to the emulators by one call to the routine SYEPUT.

GRUN : host: control of data transfer to and from the 3081/E's. emulator: call to subroutine BREAK inserted to interrupt program execution

FZOX1 FZOX2 FZOX3 FZOX4 FZOX5 FZOX6 : both on the host and the emulators these routines write into (one of) the I/O buffer(s).

XINBS XINBF : On the host, the data read is written into one of the I/O buffers On the emulator, the data are read from the I/O buffer

FFGOR : On the host, the data cards are written into one of the I/O buffers. The card controlling the event input and output is modified so that the emulator reads the initialisation structure from the buffer using GGET. The RNDM and EMUL cards are not written into the buffer. On the emulator, the data cards are read from the buffer.

Author(s) : F.G.de Bilio
Origin : GEANT3

Submitted: 01.10.84
Revised: 09.09.87

Index of Documented GEANT3 routines