CERN Program Library Long Writeup W5013

GEANT

Detector Description and Simulation Tool

Application Software Group

Computing and Networks Division

CERN Geneva, Switzerland

Requests for information should be addressed to:

CERN Program Library Office
CERN-CN Division
CH-1211 Geneva 23
Switzerland
Tel. +41 22 767 4951
Fax. +41 22 767 7155
Bitnet: CERNLIB@CERNVM
DECnet: VXCERN::CERNLIB (node 22.190)
Internet: CERNLIB@CERNVM.CERN.CH

Contact Persons: CN/ASD/SImulation section (SGIANI@CERNVM.CERN.CH)
(RAVNDAL@CERNVM.CERN.CH)

Technical Consultant: Michel Goossens /CN (GOOSSENS@CERNVM.CERN.CH)

Edition – October 1994

Chapter 1: Catalog of Geant sections

Origin :
Revision :
Documentation :

Submitted: 01.10.84
Revised: 20.04.94

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 optimise the detectors,
- develop and test the reconstruction and analysis programs, and
- interpret the experimental data.

`GEANT` is a system of detector description and simulation tools that help physicists in such studies. The `GEANT` system can be obtained from `CERN` as six Patchy [8]/CMZ [9] files: `GEANT`, `GEANG`, `GEANH`, `GEANF`, `GEANE` and `GEANX`. The program runs everywhere the CERN Program Library has been installed [1].

The `GEANT` and `GEANG` files contain most of the basic code. The `GEANH` file contains the code for the hadronic showers simulation from the program `GHEISHA` [1]. The `GEANF` file contains the source of the routines for hadronic showers development from the `FLUKA` [2, 3, 4, 5, 6, 101, 102] program which is interfaced with `GEANT` as an alternative to `GHEISHA` to simulate hadronic cascades. The `GEANE` [11] file contains a tracking package to be used, in the context of event reconstruction, for trajectory estimation and error propagation. The `GEANX` file contains the main program for the interactive version of `GEANT` (GXINT) and a few examples of application programs which may help users to get started with `GEANT`.

General information concerning `GEANT`, for example access to the source code, the list of problems and their 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 through CERN news and InterNet news group (`cern.lgeant`) and an electronic mailing list which is installed on the CERN IBM mainframe (BITnet/EARN node CERNVM). The name of the list is LGEANT (see below how to subscribe).

The first version of `GEANT` was written in 1974 as a bare framework which initially emphasised tracking of a few particles per event through relatively simple detectors. The system has been developed with some continuity over the years [12]

New versions may differ from the previous ones. Some of the modifications may lead to backward incompatibilities. The user is therefore invited to read carefully the Patch `HISTORY` of the current `GEANT` file where all changes are described in detail.

The development and the maintenance of `GEANT` are possible only thanks to the devoted and continuous collaboration of physicists around the world who use the program and contribute their feedback to the authors and maintainers at CERN. It is of course impossible to mention all of them, and new names are added frequently to the list of the contributors. The `GEANT` team wish to thank them all and expresses its hope that they will continue to help us.

`GEANT` version 3 originated from an idea of René Brun and Andy McPherson in 1982 during the development of the OPAL simulation program. GEANT3 was based on the skeleton of `GEANT` version 2 code [12].

In close collaboration with René and Andy, Pietro Zanarini developed the first versions of the graphics system as well as the early versions of the interactive package initially based on `ZCEDEX`, then upgraded to `KUIP`.

[1] At the moment of writing these are the systems on which the CERN Program Library is maintained: VM/CMS-HPO-XA-ESA, SUN, Silicon Graphics, CRAY Y-XMP, Apollo 3000 series, Apollo 10000, HP 400 series, HP 700 series, IBM RS/6000, IBM AIX/370, VAX/VMS, Alliant, Dec Ultrix, NeXT, IBM MVS-HPO-XA-ESA, Convex, MAC/MPW (partial implementation), PC-DOS, PC-Linux, PC-Windows/NT.

Glenn Patrick (RAL) implemented a first version of the electromagnetic processes. Tony Baroncelli (Roma) helped in interfacing GEANT to his hadronic shower package TATINA. Federico Carminati contributed to the interface with GHEISHA, an hadronic shower package developed by Harn Fesefeldt (Aachen).

Francis Bruyant and Michel Maire (LAPP) made substantial contributions to the geometry, tracking and physics parts of GEANT while adapting the system to the L3 environment. Francis has been, for many years, an essential collaborator, testing new ideas for the geometry and hits packages. Michel, together with Elemer Nagy and Vincenzo Innocente, developed the GEANE system.

A very important contribution to GEANT has been made by László Urbán (KFKI Budapest) who has continuously improved the electromagnetic physics package. Lazlo has spent a considerable amount of time in reading the relevant papers in the literature and in making comparisons with experimental results.

René Brun has coordinated the development and the maintenance of GEANT from 1982 until 1991 (versions 3.00 up to 3.14). Federico Carminati coordinated the development of the versions 3.15 and 3.16 between 1991 and 1993. Since January 1994, the responsability for GEANT is in the hands of Simone Giani. Before assuming this responsability, Simone made substantial improvements in the graphics and interactive packages. After he has enhanced the power of the geometry package and the performance of the tracking for a new version of GEANT: in March 1994 the version 3.21 has been released and is the current version of GEANT.

Many people contributed their work or their experience. We have tried to acknowledge their names in the manual pages and we apologise for any omissions.

Special mention should be made here of the following contributions:

S.Banerjee (contribution to the tracking package), R.Jones (contribution to the simulation of electromagnetic processes), K.Lassila-Perini (interface with FLUKA and MICAP), G.Lynch (contribution to the multiple scattering algorithms), E.Tchernyaev (original code for hidden-line removal graphics), J.Salt (original interface to the GC package).

S. Ravndal did a complete revision and update of the full documentation for the release of GEANT version 3.21.

Special thanks should go to the authors of the packages interfaced with GEANT, and in particular to Harn Fesefeldt (GHEISHA) and Alfredo Ferrari (FLUKA see later). Their patience in explaining the internals of their code, their experience and their collaborative and open attitude have been instrumental.

Another special thanks goes to Mike Metcalf, who helped to improve the English and the structure of the manual.

Any reader who is not familiar with GEANT should first have a glance at the notes numbered 001 to 009 in each section of this manual.

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

Finally, we express our thanks to Michel Goossens for translating the SCRIPT/SGML source of the original GEANT manual into LaTeX.

1 The GEANT-FLUKA interface

Since version 3.15, GEANT includes an interface with some FLUKA [2, 3, 4, 5, 6, 101, 102] routines. This part has been updated and extended in subsequent releases.

FLUKA is a standalone code with its own life. Only a few parts have been included into GEANT, namely the ones dealing with hadronic elastic and inelastic interactions.

The implementation of FLUKA routines in GEANT does not include any change, apart from interface ones and those agreed by the FLUKA authors. Whenever different options are available in FLUKA, the one suggested by the authors have been retained. Nevertheless the results obtained with FLUKA routines inside GEANT could not be representative of the full FLUKA performances, since they generally depend on other parts which are GEANT specific.

The routines made available for GEANT have been extensively tested and are reasonably robust. They usually do not represent the latest FLUKA developments, since the policy is to supply for GEANT well tested and reliable code rather than very recent developments with possibly better physics but also still undetected errors.

It is important that GEANT users are aware of the conditions at which this code has been kindly made available:

- relevant authorship and references about FLUKA [2, 3, 4, 5, 6, 101, 102] should be clearly indicated in any publication reporting results obtained with this code;
- the FLUKA authors reserve the right of publishing about the physical models they developed and implemented inside FLUKA, GEANT users are not supposed to extract from the GEANT-FLUKA code the relevant routines running them standalone for benchmarks;
- more generally, FLUKA routines contained in the GEANF file are supposed to be included and used with GEANT only: any other use must be authorised by the FLUKA authors.

2 Documentation

The main source of documentation on GEANT is this manual. Users are invited to notify any correction or suggestion to the authors.

A detailed description of the history of modifications to the GEANT code is contained in the $VERSION Patch in the GEANT file.

GEANT is part of the CERN Program Library, and problems or questions about GEANT should be directed to the Program Library Office (see next section).

GEANT problems can be submitted to an InterNet discussion group cern.lgeant.

A mailing list is maintained on CERN's central IBM machine (BITnet node CERNVM) via the LISTSERV mechanism. LISTSERV acts as a rudimentary conferencing system, which forwards the mail received to all subscribed users and to the InterNet group cern.lgeant. The list is accessible to all users who have an e-mail connection to CERNVM. To subscribe to the list from a BITnet node a user has to send the following message to LISTSERV at CERNVM using the local BITnet message facility:

```
SUBSCRIBE LGEANT
```

From a non-BITnet node, the user can send an ordinary mail message to the user LISTSERV at CERNVM containing that single line.

Together with the GEANT library comes a *correction cradle* (see section on maintenance policy) which contains the history of the modifications to the current version of the GEANT program. This file is accessible to all users in the PRO area and can be found in:

GCORR*xxx*.CAR

where *xxx* is the version number.

Documentation on the elements of the CERN Program Library used by the GEANT program is available from the CERN Program Library Office.

3 Update policy

The GEANT program is constantly updated to reflect corrections, most of the time originating from users' feed-back, and improvements to the code. This constant evolution, which is one of the reasons for the success of GEANT, poses the serious problem of managing change without disrupting stability, which is very important for physicists doing long production runs.

In the CERN Program Library maintenance scheme, three versions of any product are present at the same time on the central systems, in the OLD, PRO and NEW areas with those same names. This scheme does not apply to GEANT because every new release usually contains modifications in the physics which can produce, we hope, better but often different results with respect to the previous version. It is therefore appropriate to offer to the users an extra level of protection against running inadvertedly the wrong version by appending the version number to all the files of GEANT. In this way the users will have to change their procedure to change the version of GEANT.

On the other hand, the new user should not bother about version numbers and correction files, and so an alias is installed on all systems without any version number, always pointing to the latest released version.

As said before, users' feed-back is of paramount importance in detecting problems or areas for improvement in the system, so the new version is made available in the NEW area well before the official release. If, on the one hand, those who use this version do so at their own risk, on the other hand users are encouraged to perform as much testing as possible, in order to detect the maximum number of problems before the final release. Modifications in the pre-release version are made directly in the source code.

When problems are discovered, which may seriously affect the validity of the results of the simulation, they are corrected and the library recompiled in the /new area. To minimise network transfer for remote users and in the interest of the stability of the system, the source code of the released version in PRO is not touched, but rather the correction is applied via a so-called *correction cradle* which is a file containing the differences between the original and the corrected version in a format required by PATCHY/CMZ. Both these programs can read the original source and the correction cradle and produce the corrected source. The corrected car/cmz source and the corrected binary library are made then available in /pro at the following CERNLIB release (when /new becomes /pro).

The cmz source files contain the full history of the corrections applied with proper versioning, so that every intermediate version can be rebuild. Users at CERN should not need to use the correction cradle other than for documentation purposes. Remote users may want to obtain the cradle and apply the corrections. At every CERNLIB release the correction cradle is obviously reset to be empty, as all the corrections have been applied in the code directly. The correction cradle for the OLD version is available but has to be considered frozen. No correction is ever applied to an old version.

New versions of GEANT are moved into the PRO area synchronously with releases of the CERN Program Library. If no new version of GEANT is available at the time of the release of the Program Library, the GEANT files do not change their location and the production version remains the same.

4 Availability of the documentation

This document has been produced using LaTeX[2] with the `cernman` and `cerngeant` style options, developed at CERN. A printable version of each of the sections described in this manual can be obtained as a compressed PostScript file from CERN by anonymous ftp. You can look in the directory described in the procedure below for more details. For instance, if you want to transfer the description of the physics routines, then you can type the following (commands that you have to type are underlined):[3]

```
ftp asis01.cern.ch
Trying 128.141.201.136...
Connected to asis01.cern.ch.
220 asis01 FTP server (SunOS 4.1) ready.
Name (asis01:username): anonymous
Password: your_mailaddress
ftp> binary
ftp> cd cernlib/doc/ps.dir/geant
ftp> get phys.ps.Z
ftp> quit
```

[2] Leslie Lamport, LaTeX – *A Document Preparation System.* Addison–Wesley, 1985

[3] You can of course issue multiple `get` commands in one run.

Origin : **Submitted:** 01.10.84
Revision : **Revised:** 10.03.94
Documentation :

Introduction to the manual

The present documentation is divided into sections which follow the structure of GEANT and its major functions. Each section is identified by a *keyword* which indicates its content. Sections are in alphabetical order:

AAAA	introduction to the system;
BASE	GEANT framework and user interfaces to be read first;
CONS	particles, materials and tracking medium parameters;
DRAW	the drawing package, interfaced to HIGZ;
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 user interface;
ZZZZ	appendix.

Within each section, the principal system functions or the details of subroutines are described in a series of *papers* numbered from 001 to 999. In the upper left corner it is indicated in which Geant release the subroutines were introduced and left unchanged. The authors of the conceptual ideas or/and of the early versions of the code are acknowledged under the item **Origin**, while **Revision** contains the contributors to any important upgrade. **Documentation** is essential, but sometime implies a not negligeable amount of work. When relevant these contributions are acknowledged here. In addition all reported bugs, accepted suggestions...etc...are mentioned in the history part of the source code and correction cradle.

Subroutines which are not necessary for an understanding of the program flow and which are not intended to be called directly by the user have been omitted.

The notation [<KEYW>nnn] 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 added: 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 AAAA000. To ease access to this documentation an index appears in ZZZZ999. It gives in alphabetic order the names of all documented GEANT subroutines with references to the appropriate write up(s).

A short write up of GEANT can be obtained by collecting the papers numbered 001 to 009 in each section.

AAAA Bibliography

[1] H.J. Klein and J. Zoll. *PATCHY Reference Manual,* Program Library L400. CERN, 1988.

[2] M. Brun, R. Brun, and F. Rademakers. *CMZ - A Source Code Management System*. CodeME S.A.R.L., 1991.

[3] H.C.Fesefeldt. Simulation of hadronic showers, physics and applications. Technical Report PITHA 85-02, III Physikalisches Institut, RWTH Aachen Physikzentrum, 5100 Aachen, Germany, September 1985.

[4] P.A.Aarnio et al. Fluka user's guide. Technical Report TIS-RP-190, CERN, 1987, 1990.

[5] A.Fassò A.Ferrari J.Ranft P.R.Sala G.R.Stevenson and J.M.Zazula. FLUKA92. In *Proceedings of the Workshop on Simulating Accelerator Radiation Environments*, Santa Fe, USA, 11-15 January 1993.

[6] A.Fassò A.Ferrari J.Ranft P.R.Sala G.R.Stevenson and J.M.Zazula. A Comparison of FLUKA Simulations with measurements of Fluence and Dose in Calorimeter Structures. *Nuclear Instruments & Methods A*, 332:459, 1993.

[7] C.Birattari E.DePonti A.Esposito A.Ferrari M.Pelliccioni and M.Silari. Measurements and characterization of high energy neutron fields. *approved for pubblication in Nuclear Instruments & Methods A*.

[8] P.A.Aarnio, A.Fassò, A.Ferrari, J.-H.Moehring, J.Ranft, P.R.Sala, G.R.Stevenson and J.M.Zazula. FLUKA: hadronic benchmarks and applications. In *MC93 Int. Conf. on Monte-Carlo Simulation in High-Energy and Nuclear Physics*, Tallahassee, Florida, 22-26 February 1993. Proceedings in press.

[9] P.A.Aarnio, A.Fassò, A.Ferrari, J.-H.Moehring, J.Ranft, P.R.Sala, G.R.Stevenson and J.M.Zazula. Electron-photon transport: always so good as we think? Experience with FLUKA. In *MC93 Int. Conf. on Monte-Carlo Simulation in High-Energy and Nuclear Physics*, Tallahassee, Florida, 22-26 February 1993. Proceedings in press.

[10] A.Ferrari and P.R.Sala. A New Model for hadronic interactions at intermediate energies for the FLUKA code. In *MC93 Int. Conf. on Monte-Carlo Simulation in High-Energy and Nuclear Physics*, Tallahassee, Florida, 22-26 February 1993. Proceedings in press.

[11] V.Innocente, M.Maire, and E.Nagy. *GEANE: Average Tracking and Error Propagation Package*, July 1991.

[12] R.Brun, M.Hansroul, and J.C.Lassalle. *GEANT User's Guide*, DD/EE/82 edition, 1982.

Origin : GEANT **Submitted:** 01.10.84
Revision : **Revised:** 08.11.93
Documentation : F.Bruyant

Introduction to GEANT

1 GEANT applications

The GEANT program simulates the passage of elementary particles through the matter. Originally designed for the High Energy Physics experiments, it has today found applications also outside this domain in areas such as medical and biological sciences, radio-protection and astronautics.

The principal applications of GEANT in High Energy Physics are:

- the *transport* of particles (*tracking* in this manual) through an experimental setup for the simulation of detector response;
- the graphical representation of the setup and of the particle trajectories.

The two functions are combined in the interactive version of GEANT. This is very useful, since the direct observation of what happens to a particle inside the detector makes the debugging easier and may reveal possible weakness of the setup (also sometimes of the program!).

In view of these applications, the GEANT system allows you to:

- describe an experimental setup by a structure of geometrical volumes. A MEDIUM number is assigned to each volume by the user ([GEOM]). Different volumes may have the same medium number. A medium is defined by the so-called TRACKING MEDIUM parameters, which include reference to the MATERIAL filling the volume [CONS];
- accept events simulated by Monte Carlo generators [KINE];
- transport particles through the various regions of the setup, taking into account geometrical volume boundaries and physical effects according to the nature of the particles themselves, their interactions with matter and the magnetic field [TRAK], [PHYS];
- record particle trajectories and the response of the sensitive detectors [TRAK], [HITS];
- visualise the detectors and the particle trajectories [DRAW], [XINT].

The program contains *dummy* and *default* user subroutines called whenever application-dependent actions are expected.

It is the responsibility of the user to:

- code the relevant user subroutines providing the data describing the experimental environment;
- assemble the appropriate program segments and utilities into an executable program;
- compose the appropriate data records which control the execution of the program.

The section [BASE] of this manual gives more information on the above.

Note: the names of the dummy or default user subroutines have GU or UG as their first two letters.

2 Event simulation framework

The framework offered by GEANT for event simulation is described in the following paragraphs, in order to familiarise the reader with the areas where user interventions are expected. For each item we will indicate in brackets the relevant section where more information can be found.

At the same time, the GEANT data structures are introduced. This is important as the coding to be provided by the user most often consists of storing and retrieving information from data structures, or reading or writing data structures. For simple applications user routines are provided as an interface to the data structures partially hiding them from the users. For advanced users of GEANT, some idea of the layout of the data in memory is helpful. GEANT data structures are logically related set of data which are physically stored in the /GCBANK/ common block. The position of each structure is contained in an INTEGER variable which is constantly kept up-to-date by ZEBRA. By convention the names of these variable, called *pointers* begin with J, and they are used in this manual to designate the structure they point to.

A main program has to be provided by the user ([BASE100]) for *batch* type operation. For interactive operation a main program is provided, both binary and source, in the library directory both at CERN and in the standard distribution tape of the CERN Program Library. The file is called gxint<ver>.<ext>, where <ver> is the version of GEANT to which this file belongs and <ext> is the system-dependent file-name extension to denote a FORTRAN source or an object file. This file should be loaded in front of all other files when assembling a GEANT application. The source is provided in case the user wants to modify it, in particular changing the size of the commons /GCBANK/ or /PAWC/.

The main program allocates the dynamic memory for ZEBRA and HBOOK and passes control to the three phases of the run:

1. initialisation
2. event processing
3. termination

where in each of the three phases the user can implement his own code in the appropriate routines.

3 Initialisation

The initialisation is controlled by the user in the subroutine UGINIT who has the responsibility to call the appropriate routines ([BASE100]). It consists of the following steps, most of them performed through calls to GEANT subroutines:

GINIT	initialise the GEANT common blocks with default values ([ZZZZ010], [BASE110]);
GFFGO	read *free format* data records either to modify the default options ([BASE040], [BASE110]) or to provide information on the current run;
GZINIT	initialise the memory manager, the link areas and the run header bank JRUNG ([BASE110]);
GDINIT	initialise the drawing package ([DRAW]), calling this routine without having initialised the graphics package via a call to IGINIT or HPLINT will cause GEANT to abort;
GPART	(or GSPART) fill the data structure JPART with the particle properties ([CONS]);
GMATE	(or GSMATE) fill the data structure JMATE with the characteristics of the materials used ([CONS]);
User code	it is the responsibility of the user to:

- define the geometry of the different components of the setup ([GEOM]), stored in the JROTM and JVOLUM data structures;
- define tracking medium parameters ([CONS], [TRAK]), stored in the JTMED data structure;

- specify which elements of the geometrical setup should be considered as *sensitive detectors*, giving a *response* when hit by a particle ([HITS]);
- usually all done in a user routine called UGEOM;

GGCLOS process all the geometrical information provided by the user and prepare for particle transport;

GBHSTA book standard GEANT histograms if required by the user with the data record HSTA ([BASE040], [BASE110]);

GPHYSI compute energy loss and cross-section tables and 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 initialise event processing and create the event header bank JHEAD;

GTRIG process one event;

GTRIGC clean up the portion of memory used by the event and check that enough time is left for the next event ([BASE200]).

GTRIG calls the following user routines:

GUKINE generates or reads ([IOPA]) the kinematics of the event and stores it in the data structures JVERTX and JKINE ([KINE]);

GUTREV calling GTREVE which performs the following operations for each vertex in turn:

1. moves all the particles attached to the vertex from the permanent stack JKINE to the temporary stack JSTAK;
2. controls the propagation of each particle though the setup by calling GUTRAK/GTRACK ([TRAK]); each particle is tracked in turn and when a sensitive detector is traversed, the user may store any useful information in the data structure JHITS via the routines described in the section [HITS];

The JSTAK data structure is a LIFO (Last In – First Out) stack. Secondary products generated by the current particle transported are processed before proceeding to the next particle. It is very important to understand that by default GEANT does not follow the secondary particles generated. It is the responsibility of the user to indicate which particles should be followed via the routines GSKING/GSKPHO.
The data structure JXYZ, containing the coordinates of space points along the tracks, can be filled by the user during tracking ([TRAK]).

GUDIGI simulate the detector responses for the event, using the information recorded in the data structure JHITS during particle transport, and store the results in the data structure JDIGI ([HITS]);

GUOUT perform all the processing at the end of the event and output the relevant data structures ([IOPA]).

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

- hadronic interactions can be simulated via either the GHEISHA [1] or FLUKA [2, 3, 4, 5, 6] hadronic shower generator. In the subroutines GUPHAD and GUHADR ([TRAK]) the user may select the hadronic shower generation program to be used. The default for GEANT is GHEISHA;
- after each tracking step along the track, 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 transferring a secondary product either in the stack JSTAK or in the event structure JVERTX/JKINE via the subroutine GSKING. In the subroutine GSSTAK, called by GSKING, a user routine GUSKIP is called which permits skipping any unwanted track before entering it in the stack for subsequent transport;

- the subroutine GUSWIM is called by the the routines which transport charged particles when in a magnetic field; it selects and calls the appropriate routine to transport the particle. Although formally a user routine, the default version provided by GEANT is usually appropriate for most situations. The magnetic field, unless it is constant along the Z axis, has to be described via the subroutine GUFLD.

5 Termination

The termination phase is under the control of the user ([BASE300]) via the routine GULAST. In simple cases it may consist of a call to the subroutine GLAST which computes and prints some statistical information (time per event, use of dynamic memory, etc.).

Origin :
Revision :
Documentation : F.Bruyant, S.Ravndal

Submitted: 01.10.84
Revised: 19.10.94

Simplified Program Flow Chart

MAIN	user routine
GZEBRA	initialisation of ZEBRA system, dynamic core allocation
UGINIT	user routine
GINIT	initialisation of GEANT variables
GFFGO	interpretation of data records
GZINIT	initialisation of ZEBRA core divisions and link areas
GPART/GSPART	creation of the *particle* data structure JPART
GMATE/GSMATE	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
GFINDS	find current volume in the geometry tree
GUSTEP	(user) recording of hits in data structure JHITS and of space points in data structure JXYZ
GUPARA	called if the particle falls below the tracking threshold
GTGAMA/GTELEC/...	tracking of particle according to type
GFSTAT	fill banks for volume statistics
GSTRAC	store information of the current track segment
GUSTEP	(user) recording of hits in data structure JHITS and of space points in data structure JXYZ
GTMEDI	finds in which volume/medium the current space point is
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 GEANT termination

Origin : **Submitted:** 01.10.84
Revision : **Revised:** 20.03.94
Documentation : M.Maire

The data structures and their relationship

1 Dynamic memory

The GEANT data structures are stored in the common /GCBANK/ accessible through the following Patchy sequence: The /GCLINK/ variables are pointers to the GEANT data structures in the /GCBANK/ common. They belong to a permanent area declared in GZINIT.

```
+SEQ /GCBANK/  Dynamic core for the GEANT data structures
```

```
      PARAMETER (KWBANK=69000,KWWORK=5200)
      COMMON/GCBANK/NZEBRA,GVERSN,ZVERSN,IXSTOR,IXDIV,IXCONS,FENDQ(16)
     +             ,LMAIN,LR1,WS(KWBANK)
      DIMENSION IQ(2),Q(2),LQ(8000),IWS(2)
      EQUIVALENCE (Q(1),IQ(1),LQ(9)),(LQ(1),LMAIN),(IWS(1),WS(1))
      EQUIVALENCE (JCG,JGSTAT)
      COMMON/GCLINK/JDIGI ,JDRAW ,JHEAD ,JHITS ,JKINE ,JMATE ,JPART
     +      ,JROTM ,JRUNG ,JSET  ,JSTAK ,JGSTAT,JTMED ,JTRACK,JVERTX
     +      ,JVOLUM,JXYZ  ,JGPAR ,JGPAR2,JSKLT
C
```

The figure on the next page indicates the ralations between the GEANT data structures. Detailed description of the data structure parts can be found in the following sections:

JRUNG	[BASE299]				
JPART	[CONS399]	JMATE	[CONS199]		
JROTM	[GEOM299]	JTMED	[CONS299]		
JVOLUM	[GEOM199]				
JSET	[HITS199]	JDRAW	[DRAW399]		
JHEAD	[BASE299]	JKINE	[KINE199]	JVERTX	[KINE199]
JSTAK	[TRAK399]				
IDIGI	[HITS399]	JHITS	[HITS299]	JXYZ	[TRAK 499]

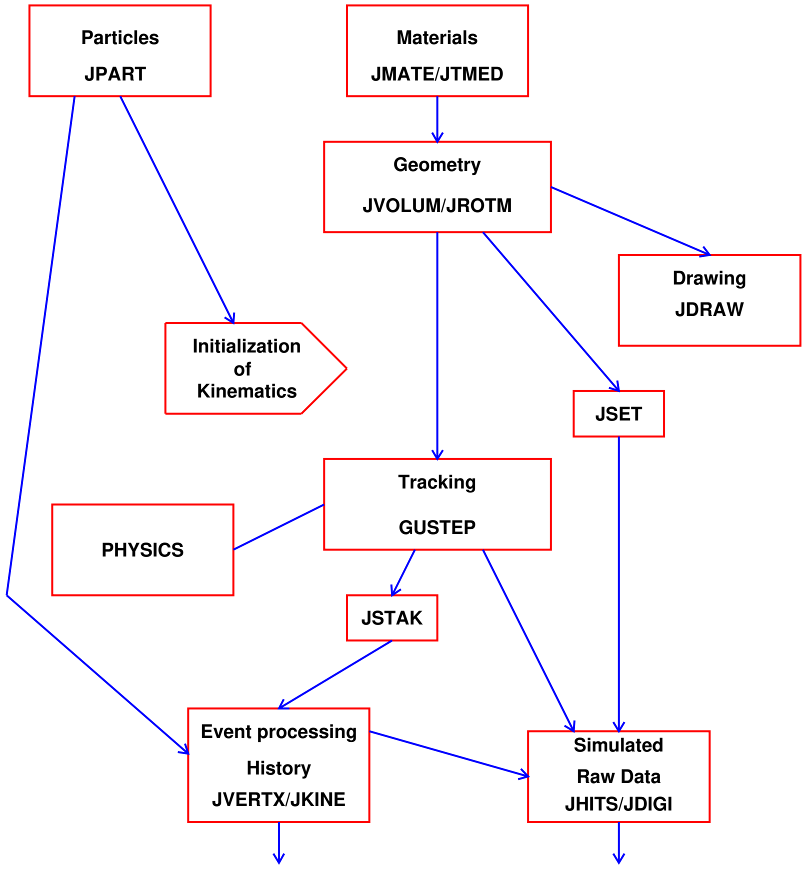


Figure 1: Relation between GEANT data structures

2 Common blocks

The communication between program segments of the GEANT system is assured 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 actual arguments and through the common block variables. A detailed list of the user accessed common blocks is given in [ZZZZ010]. Their also the variables initialized in GINIT and the possibility in overriding them through data records [BASE040] or interactive commands [XINT] are specified.

In most of the cases there is a correspondence between a given data structure and a given common block where the current contents of the banks are stored. The labelled common blocks are accessible through Patchy/CMZ sequences identified by the name of the COMMON. They are defined in the Patch GCDES.

Note:

Unless otherwise specified, the long range variables are initialised in GINIT. When non-zero, default values are quoted between brackets. If the value may be modified the keyword for the data record and for the interactive command is also given in bold characters between brackets.

Origin :
Revision :
Documentation : F.Bruyant, M.Maire

Submitted: 01.10.84
Revised: 16.12.93

Summary of Data Records

1 Introduction

GEANT uses the FFREAD [7] package to read *free format* data records in the routine GFFGO. The keywords accepted by GFFGO can be classified as:

1. general control of the run;
2. control of the physics processes;
3. debug and I/O operations;
4. user applications;
5. Lund event generation.

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

KEY keyword, any number of characters truncated to the first 4 unless otherwise specified by the user;
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:

- variable FORTRAN name;
- short description (more detail in [ZZZZ010]);
- labelled common where it is stored;
- default value, usually from GINIT.

When a record is decoded, the values entered by the user in free format 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 agree with the type of the corresponding variable.
For example the data record:

```
RUNG   5   201
```

presets the run and event number to 5 and 201 respectively. None of the records mentioned below is mandatory.

2 User defined data records

Before calling GFFGO the user may define private data records 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 GEANT pre-defined records.

3 Summary of GEANT data records

3.1 General control

KEY	N	I	VAR	Short description	COMMON	GINIT
HSTA	20	M	LHSTA	names of required standard histograms, see [BASE110]	/GCLIST/	Blank
OPTI	1	I	IOPTI	automatic optimisation of the geometry via GSORD	/GCOPTI/	1
RNDM	2	I	NRNDM	initial random number seed (2 words)	/GCFLAG/	0
RUNG	2	I	IDRUN	user run number	/GCFLAG/	1
			IDEVT	first user event number	/GCFLAG/	1
SORD	1	I	ISTORD	stack ordering flag	/GCSTAK/	0
TRIG	1	I	NEVENT	total number of events to process	/GCFLAG/	10000000
TIME	3	M	TIMINT	time left after initialisation (see **Note below**)	/GCTIME/	
			TIMEND	time required for termination	/GCTIME/	1
			ITIME	test every ITIME events	/GCTIME/	1

Note: the time allowed for the job after initialisation cannot be set by the user via the data record. To set the total time for the job the user should call the TIMEST routine at the beginning of the program before any call to GEANT routines. This variable in the data record has not been removed for backward compatibility.

3.2 Control of physics processes

For more information on the use of these flags, see [PHYS001].

KEY	N	I	VAR	Short description	COMMON	GINIT
ANNI	1	I	IANNI	annihilation	/GCPHYS/	1
AUTO	1	I	IGAUTO	automatic computation of the tracking medium parameters	/GCTRAK/	1
BREM	1	I	IBREM	bremsstrahlung	/GCPHYS/	1
CKOV	1	I	ICKOV	Čerenkov photon generation	/GCTMED/	0
COMP	1	I	ICOMP	Compton scattering	/GCPHYS/	1
CUTS	16	R	Kinetic energy cuts in GeV:			
			CUTGAM	cut for for gammas	/GCCUTS/	0.001
			CUTELE	cut for electrons	/GCCUTS/	0.001
			CUTNEU	cut for neutral hadrons	/GCCUTS/	0.01
			CUTHAD	cut for charged hadrons	/GCCUTS/	0.01
			CUTMUO	cut for muons	/GCCUTS/	0.01
			BCUTE	cut for electron bremsstrahlung	/GCCUTS/	GUTGAM
			BCUTM	cut for muon and hadron bremsstrahlung	/GCCUTS/	CUTGAM
			DCUTE	cut for δ-rays by electrons	/GCCUTS/	10^4
			DCUTM	cut for δ-rays by muons	/GCCUTS/	10^4
			PPCUTM	total energy cut for direct pair production by muons	/GCCUTS/	0.01
			TOFMAX	time of flight cut in seconds	/GCCUTS/	10^{10}
			GCUTS	5 user words	/GCCUTS/	0
DCAY	1	I	IDCAY	decay	/GCPHYS/	1
DRAY	1	I	IDRAY	δ-ray	/GCPHYS/	1
KEY	N	I	VAR	Short description	COMMON	GINIT
ERAN	3	M	cross-section tables structure:			
		R	EKMIN	minimum energy for the cross-section tables	/GCMULO/	10^{-5}
		R	EKMAX	maximum energy for the cross-section tables	/GCMULO/	10^4
		I	NEKBIN	number of logarithmic bins for cross-section tables	/GCMULO/	90
HADR	1	I	IHADR	hadronic process	/GCPHYS/	1
LABS	1	I	ILABS	Čerenkov light absorbtion	/GCPHYS/	0
LOSS	1	I	ILOSS	energy loss	/GCPHYS/	2
MULS	1	I	IMULS	multiple scattering	/GCPHYS/	1
MUNU	1	I	IMUNU	muon nuclear interaction	/GCPHYS/	1
PAIR	1	I	IPAIR	pair production	/GCPHYS/	1
PFIS	1	I	IPFIS	photofission	/GCPHYS/	0
PHOT	1	I	IPHOT	photo electric effect	/GCPHYS/	1
RAYL	1	I	IRAYL	Rayleigh scattering	/GCPHYS/	0
STRA	1	I	ISTRA	energy fluctuation model	/GCPHYS/	0
SYNC	1	I	ISYNC	synchrotron radiation generation	/GCPHYS/	0

3.3 Debug and I/O operations

KEY	N	I	VAR	Short description	COMMON	GINIT
DEBU	3	M	IDEMIN	first event to debug.	/GCFLAG/	0
			IDEMAX	last event to debug	/GCFLAG/	0
			ITEST	print control frequency	/GCFLAG/	0
GET	20	M	LGET	NGET names of data structures to fetch (see **Note**)	/GCLIST/	Blank
PRIN	20	M	LPRIN	NPRIN user keywords to print data structure (see **Note**)	/GCLIST/	Blank
RGET	20	M	LRGET	NRGET names of data structures to fetch from RZ files (see **Note**)	/GCRZ/	Blank
RSAV	20	M	LRSAVE	NRSAVE names of data structures to save from RZ files (see **Note**)	/GCRZ/	Blank
SAVE	20	M	LSAVE	NSAVE names of data structures to save (see **Note**)	/GCLIST/	Blank
SWIT	10	I	ISWIT	user flags for debug	/GCFLAG/	0

Note: the user data records for I/O have no effect on the GEANT system, and the user is supposed to analyse them at run time and take corresponding action. For instance, a use of the PRIN data record could be the following:

```
CALL GLOOK('VOLU',LPRIN,NPRIN,IPRES)
IF(IPRES.NE.0) THEN
   CALL GPVOLU(0)
ENDIF
```

All the names quoted here are given as 4-character strings in input and their ASCII equivalent is read into the corresponding variable. The same applies to the user lists of the following section.

3.4 User applications

KEY	N	I	VAR	Short description	COMMON	GINIT
KINE	11	M	IKINE	user flag	/GCKINE/	0
			PKINE	10 user words	/GCKINE/	10^{10}
SETS	20	M	LSETS	user words for detector sets	/GCLIST/	Blank
STAT	20	M	LSTAT	user words to control statistics	/GCLIST/	Blank
PLOT	20	M	LPLOT	user words to control plots	/GCLIST/	Blank
GEOM	20	M	LGEOM	user words to control geometry setup	/GCLIST/	Blank
VIEW	20	M	LVIEW	user words to control view banks	/GCLIST/	Blank

See note in the previous section on the use of these data records.

3.5 Lund event generation

KEY	N	I	VAR	Short description	COMMON	GINIT
LUND	2	M	IFLUND	flavour code (See /GLUNDI/)	/GCLUND/	0
		R	ECLUND	total CMS energy	/GCLUND/	94

3.6 Scan geometry control

The scan geometry has been introduced in GEANT version 3.15 with the idea of developing a general parametrisation scheme. While the scan geometry can already be built, the parametrisation scheme has not yet been developed, so the following data records have to be considered as a *forward compatibility* feature.

KEY	N	I	VAR	Short description	COMMON	GINIT
PCUT	6	M	parametrisation control			
			IPARAM	parametrisation flag	/GCPARM/	0
			PCUTGA	threshold for gammas	/GCPARM/	0
			PCUTEL	threshold for e^-e^+	/GCPARM/	0
			PCUTNE	threshold for neutral hadrons	/GCPARM/	0
			PCUTHA	threshold for charged hadrons	/GCPARM/	0
			PCUTMU	threshold for muons	/GCPARM/	0
PNUM	2	I	parametrisation stack size			
			MPSTAK	size of the parametrisation stack	/GCPARM/	1000
			NPGENE	number of pseudo-particles generated	/GCPARM/	20
SCAN	8	M	scan geometry control			
			SCANFL	scan processing flag (LOGICAL)	/GCSCAN/	.FALSE.
			NPHI	number of divisions in ϕ	/GCSCAN/	90
			PHIMIN	minimum value of ϕ	/GCSCAN/	0
			PHIMAX	maximum value of ϕ	/GCSCAN/	360
			NTETA	number of divisions in θ	/GCSCAN/	90
			TETMIN	minimum value of θ	/GCSCAN/	0
			TETMAX	maximum value of θ	/GCSCAN/	180
			MODTET	type of θ division	/GCSCAN/	1
SCAL	32	M	ISLIST	list of volumes to be *scanned*	/GCSCAN/	Blank

KEY	N	I	VAR	Short description	COMMON	GINIT
SCAP	6	R	scan parameters			
			VX	scan vertex X coordinate	/GCSCAN/	0
			VY	scan vertex Y coordinate	/GCSCAN/	0
			VZ	scan vertex Z coordinate	/GCSCAN/	0
			FACTX0	scale factor for SX0	/GCSCAN/	100
			FACTL	scale factor for SL	/GCSCAN/	1000
			FACTR	scale factor for R	/GCSCAN/	100

3.7 Landau fluctuations versus δ-rays

In order to avoid double counting between energy loss fluctuations (ILOSS=2) and generation of δ-rays IDRAY=1, if ILOSS = 2 the default value for δ-ray generation is set to 0 and it cannot be changed. The different cases are summarised in the table below.

	Full fluctuations ILOSS = 2 (D)	Restricted fluctuations ILOSS = 1 or 3	No fluctuations ILOSS = 4
IDRAY	0	1	1
DCUTE	10 TeV	CUTELE	CUTELE
DCUTM	10 TeV	CUTELE	CUTELE

Origin :
Revision :
Documentation : R.Brun, F.Bruyant

Submitted: 01.10.84
Revised: 26.10.93

The reference systems and physical units

1 The MAster Reference System (MARS)

The kinematic variables of the particles transporter by GEANT are always referred to the so-called MAster Reference System (MARS). This system is implicitly defined as the local reference system of the first volume defined, which contains all the others. This is a Cartesian coordinate system with axis $\hat{x}, \hat{y}, \hat{z}$ where $\hat{z} = \hat{x} \times \hat{y}$. If the axes are labelled (X,Y,Z), then the point P is represented in fig 2.

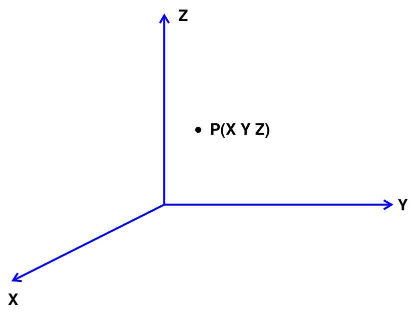


Figure 2: GEANT reference system

Tracking is performed in the MARS and the input position for user routines such as the magnetic field routine is given in this system.

2 The local reference systems (MRS and DRS)

As explained in [GEOM001], the setup is described via the definition of an initial volume inside which all the others will be positioned. In GEANT terminology, each time a volume has contents, created either via division or by positioning other volumes inside, it is called a MOTHER. The volumes contained are called DAUGHTERs, and they, in turn, can contain volumes to a depth of 15 levels. This is sometimes referred to as a *Russian doll* geometry.

Every volume defined in GEANT has a reference system attached to it (see GEOM section). When this volume has contents, this is referred to as the Mother Reference System (MRS, with origin in O_m). Daughters are positioned inside the mother with respect to the MRS. The MRS of the first volume defined, containing all the others, is nothing else than the MARS.

Each one of the daughters has its own reference system, which is referred to as the Daughter Reference System, or DRS with origin in O_d.

The transformation of a point from the MRS (V_m) to the DRS (V_d), at any level, is performed using a rotation matrix $[R]$ and a translation vector T via the relation :

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

The components of T are the projections of the vector (O_m, O_d) onto the MRS axes. The rotation matrices are computed from the spherical angles of each of the axes of the daughter reference systems (I, II, III) with respect to the mother reference system (1, 2, 3). The spherical angles Θ and Φ of a direction D are defined as follows :

Θ is the angle formed by the axis 3 and D ($0° < \Theta < 180°$).

Φ is the angle formed by the axis 1 and the projection of D onto the plane defined by the axes 1 and 2 ($0° < \Phi < 360°$).

Examples are given in [GEOM200]. The various rotation matrices required for a given setup must be defined by the user during the initialisation stage. A number is assigned to each matrix [GEOM200]. The translation vector and the number of the rotation matrix are specified by the user when the volumes are positioned inside their mother [GEOM110].

3 Physical units

Unless otherwise specified, the following units are used throughout the program: centimeter, second, kilogauss, GeV, GeV c^{-1} (momentum), GeV c^{-2} (mass) and degree.

Origin :
Revision :
Documentation : R.Brun, S.Ravndal

Submitted: 01.10.84
Revised: 10.03.94

Examples of GEANT application

This section contains a skeleton of a standard user program GEXAMP to use the GEANT system. More detailed examples can be found in the standard examples GEXAM1 - 6. The recommended user routines are indicated in bold characters and will be explained more in detail in the following.

```
      PROGRAM GEXAMP
      PARAMETER (NGBANK=50000, NHBOOK=5000)
      COMMON/GCBANK/Q(NGBANK)
      COMMON/PAWC  /H(NHBOOK)
C-->  Initialises  HBOOK and  GEANT memory
      CALL GZEBRA( NGBANK)
      CALL HLIMIT(-NHBOOK)
C-->  Open graphics system
      CALL HPLINT(0)
      CALL IGMETA(8,0)
C-->   GEANT initialisation
      CALL  UGINIT
C-->  Start events processing
      CALL GRUN
C-->  End of Run
      CALL  UGLAST
      END
*-----------------------------------------------
      SUBROUTINE  UGINIT
C-->  Initialise  GEANT
      CALL GINIT
C-->  Read data records
      OPEN(4,FILE='gcards.dat',STATUS='UNKNOWN')
      CALL GFFGO
C-->  Initialise data structure
      CALL GZINIT
C-->  Initialise graphics
      CALL GDINIT
      IF(NRGET.GT.0) THEN
C-->  Read data structures from file
         CALL GRFILE(1,'mygeom.dat','I')
      ELSE
C-->  Particle table initialisation
         CALL GPART
C-->  Geometry and materials description
         CALL UGEOM
      ENDIF
C-->  Energy loss and cross-sections tables
      CALL GPHYSI
      IF(NRSAVE .GT. 0) THEN
C-->  Save permanent data structures
         CALL GRFILE(2,'mysave.dat','NO')
      ENDIF
C-->  Print banks
      CALL GPRINT('MATE',0)
      CALL GPRINT('TMED',0)
      CALL GPRINT('VOLU',0)
C-->  Book histograms
      END
*-----------------------------------------------
      SUBROUTINE  UGEOM
C-->  Defines material, tracking media
C-->  and geometry.
C-->  Close geometry banks.
      CALL GGCLOS
      END
*-----------------------------------------------
      SUBROUTINE  GUKINE
C-->  Generates kinematics
C-->  Data card  KINE itype x y z px py pz
+SEQ, GCFLAG.
+SEQ, GCKINE.
      CALL GSVERT(PKINE,0,0,0,0,NVERT)
      CALL GSKINE(PKINE(4),IKINE,NVERT,0,0,NT)
C-->  Print kinematic
      IF (IDEBUG.NE.0) THEN
         CALL GPRINT('VERT',0)
         CALL GPRINT('KINE',0)
      END IF
      END
*-----------------------------------------------
      SUBROUTINE  GUSTEP
C-->  Called at the end of each tracking
C-->  step.
+SEQ, GCKINE.
C-->  Debug event
      CALL GDEBUG
C-->  Store the created particles
      IF (NGKINE.GT.0) CALL GSKING (0)
      END
*-----------------------------------------------
      SUBROUTINE  UGLAST
C-->  Termination routine
C-->  Print histograms and statistics
      CALL GLAST
C-->  Close  HIGZ/GKS file
      CALL IGEND
      END
```

1 Notes

- Whenever possible GEANT makes use of the ZEBRA store for large data structures. This allows it to adapt the size of the program data portion to the size of the problem. The total amount of space required depends on the application. GEANT can run with as little as 50,000 words or less, but for large detectors it is not uncommon to declare stores of several million words. The call to GZEBRA initialises the common /GCBANK/ to receive the GEANT data structures. This call is necessary before any other routine of the GEANT system is called.

- The call to HLIMIT initialises the ZEBRA system to use the /PAWC/ common block for the HBOOK histogram package. The size of the common depends on the number and size of the plot requested. The ZEBRA system must be initialised only once, and the negative argument to HLIMIT prevents a second initialisation of the system. The HLIMIT call has to be placed **after** the call to GZEBRA and the argument has to be the dimension of the /PAWC/ common block with a negative sign in front.
- The main program is intended for *batch* applications, while to run the simulation interactively, the interactive main program called GXINT should be linked in front of the user code.
- The program shown will require the graphic libraries in the link sequence. Often, for batch production or for small tests, graphics is not needed, and not loading the graphics code makes the program smaller. To avoid loading graphic routines the calls to IGINIT, IGMETA, IGEND, GDINIT and GDEBUG should be removed.

  If, on the other hand, the user is interested in including the routine GDEBUG and in excluding graphics at the same time, then the following routine should be included in the code:

```
      SUBROUTINE GDCXYZ
      ENTRY IGSA
      ENTRY GDTRAK
      END
```

  which will avoid every reference to the graphics routines from GDEBUG.
- The user code to define the *tracking media* and the geometry of the setup should be inside the routine UGEOM. The pre-initialised data structured can be read from disk, but it is recommended to call GPHYSI in any case, to initialise the cross-section tables. An example of a full material, geometry and detector design is given below and has been extracted from the example GEXAM3. Here only major calls are shown, the redundant parts can be found in the source code of UGEOM in GEXAM3.

  The example shows the basic concept in GEANT. First material parameters are defining the properities of a detector material calling the subroutine GSMATE. Here in addition to the 16 predefined materials, the material definition of Calcium is examplary shown. More information towards the predefined materials and further use of material definition routines can be found in the section CONS001 - CONS101. Then tracking parameters are associated to the materials, defining a so called tracking medium. Each GEANT volume must be associated to an existing tracking medium. Here in the example the tracking medium 'TARGET' is defined to exist of Calcium.

  In the example shown below several detector volumes are defined using the subroutine GSVOLU. The defined volume have associated parameters of name, shape, tracking medium and shape parameters. In this example the volume 'TGT ' consists of the previously defined tracking medium 'TARGET'.The volumes (and if necessary identical copies of them) are then positioned according to the detector geometry. The volumes are positioned on the same level, or inside each other. By setting the parameter ONLY or MANY in the call of GSPOS the user has the opportunity to tell either GEANT the logical volume structure or to apply boolean operations (cutting, joining and intersection) between two positioned volumes. More information about the concept in defining volumes and positioning can be retrieved from the section GEOM.

  Finally the user is required to classify into sets all sensitive detectors (defined as those volume defined as detector via GSDET and other related routines, for which he wants to store hits in the hit data structure JHITS.

```
      SUBROUTINE  UGEOM
+SEQ,GCLIST
+SEQ,GCONSP
      COMMON/DLSFLD/ISWFLD,FLDVAL
C-->  Defining material parameters
C-->  Defining geometry parameters
C-->  Defining positioning parameters
C-->  Data statements, left out here, to
C-->  Define materials and mixtures
      CALL GSMATE(17,'CALCIUM$',
     + 40.08,20.,1.55,10.4,23.2,0,0)
C-->  .......
C-->  further material an mixture definitions
C-->  .......
C-->  Defining tracking media
      CALL GSTMED( 2,'TARGET    $',
     + 17,0,0,0.,10.,.2,.1,.001,.5,0,0)
C-->  .......
C-->  defining further media
C-->  .......
C-->  Define the reference frame
      CALL GSVOLU
     +     ('CAVE','BOX ',1,CAVPAR,3,ICAVE)
C-->  The target box is defined
      CALL GSVOLU
     +     ('TGT ','BOX ',2,TGTPAR,3,ITGT )
      CALL GSVOLU
     +     ('TBIN','TRD1',3,TBIPAR,4,ITBIN)
      CALL GSVOLU
     +     ('TBOU','TRD1',4,TBOPAR,4,ITBOU)
      CALL GSVOLU
     +     ('ARM ','TRD1',1,ARMPAR,4,IARM)
      CALL GSVOLU
     +     ('FDIN','BOX ',9,FDIPAR,3,IFDIN)
      CALL GSVOLU
     +     ('FDOU','BOX ',4,FDOPAR,3,IFDOU)
C-->  Define drift wire planes
      CALL GSVOLU
     +     ('FSP ','BOX ',13,FDIPAR,3,IFSP)
C-->  .......
C-->  further geometry definitions
C-->  .......
C-->  Positioning the daughter volumes in
C-->  their mother volume.
```

```
      CALL GSPOS
     + ('TGT ',1,'TBIN', 0., 0.,-5.08,0,'ONLY')
      CALL GSPOS
     + ('TGT ',2,'TBIN', 0., 0.,-2.54,0,'ONLY')
      CALL GSPOS
     + ('TGT ',3,'TBIN', 0., 0., 0.  ,0,'ONLY')
      CALL GSPOS
     + ('TGT ',4,'TBIN', 0., 0., 2.54,0,'ONLY')
      CALL GSPOS
     + ('TGT ',5,'TBIN', 0., 0., 5.08,0,'ONLY')
      CALL GSPOS
     + ('TBIN',1,'TBOU', 0., 0.,   0.,0,'ONLY')
      CALL GSPOS
     + ('TBOU',1,'CAVE', 0., 0.,  ZTG,0,'ONLY')
      CALL GSPOS
     + ('ARM ',1,'CAVE',XLARM,0.,ZLARM,1,'ONLY')
      CALL GSPOS
     + ('ARM ',2,'CAVE',XRARM,0.,ZRARM,2,'ONLY')
      CALL GSPOS
     + ('FDOU',1,'ARM ',0.,0., DFDO  ,0,'ONLY')
      CALL GSPOS
     + ('FDIN',1,'FDOU',0.,0., 0.    ,0,'ONLY')
      CALL GSPOS
     + ('FSP ',1,'FDIN',0.,0.,-2.9975,0,'ONLY')
C-->  .......
```

```
C-->  positioning of further volumes
C-->  .......
C-->  Print the stored definitions
      CALL GLOOK('VOLU',LPRIN,NPRIN,ILOOK)
      IF(ILOOK.NE.0) CALL GPVOLU(0)
      CALL GLOOK('ROTM',LPRIN,NPRIN,ILOOK)
      IF(ILOOK.NE.0) CALL GPROTM(0)
      CALL GLOOK('TMED',LPRIN,NPRIN,ILOOK)
      IF(ILOOK.NE.0) CALL GPTMED(0)
      CALL GLOOK('MATE',LPRIN,NPRIN,ILOOK)
      IF(ILOOK.NE.0) CALL GPMATE(0)
      CALL GLOOK('PART',LPRIN,NPRIN,ILOOK)
      IF(ILOOK.NE.0) CALL GPPART(0)
C-->  Clean up volume banks and perform
C-->  optimization
      CALL GGCLOS
C-->  Define sensitive detector parts
      CALL GSDET
     &('DRFT','FSP ',2,NAFD ,NBITSV,1,100,
     &100,IDRFT,IFD )
C-->  Define hit parameters
      CALL GSDETH('DRFT','FSP ',9,NAMESH,
     &NBITSH,ORIG,FACT)
      RETURN
      END
```

- It is convenient to store the input data records (see [BASE040]) in an auxiliary file (`gcards.dat` in the example). This allows to have a standard input file and to overwrite selected input data records as needed. If, for instance, the standard `gcards.dat` file contains the record `TRIG 1000` and a short test run is requested this can be obtained with the following input:

  ```
  READ 4
  TRIG 10
  STOP
  ```

  the first line instructs `FFREAD` to open and process the file connected with logical unit 4, and the second line (re-)defines the number of events to be processed. The `STOP` command ends the `FFREAD` processing of the input.

- In the above example the common blocks have not been expanded in the code. The notation used is the one of the `PATCHY/CMZ` [8, 9] code management systems. These products, among other things, can run as pre-processors, replacing the `+SEQ,...` instructions with the corresponding code fragments. Users are strongly recommended to use these systems to include `GEANT` common blocks in their code.

  Long experience in supporting `GEANT` users has shown that, as the user program grows, typing errors in the insertion of the common blocks *by hand* become very common, but difficult to find. The investment needed to learn a code management system at the user level is usually negligible compared with the time and energy needed in hunting a problem introduced by a mistyped common.

Origin : R.Brun
Revision :
Documentation :

Submitted: 01.06.83
Revised: 16.12.93

The system initialisation routines

CALL **GZEBRA** (NZ)

Initialises the ZEBRA memory manager to use the common /GCBANK/ to store the GEANT data structures.

NZ (INTEGER) size of the /GCBANK/ common as it is dimensioned in the main program.

The size of the dynamic memory is set to NZ-30. The common /GCBANK/ must be dimensioned in the main program. ZEBRA [8] must be initialised only once. The call to the HBOOK initialisation routine HLIMIT tries to initialise ZEBRA as well, and this will cause a program abort. To avoid this, HLIMIT must be called **after** GZEBRA and its argument must be a negative number whose absolute value is the size of the /PAWC/ common containing the histograms. This is shown in the example of main program given in [BASE100].

CALL **GINIT**

Presets labelled common block variables to default values. See [ZZZZ010] for more information.

CALL **GFFGO**

Reads a set of data records via the FFREAD package. See [BASE040] for more information on the possible data records. GFFGO must be called after GINIT.

CALL **GZINIT**

Initialises the ZEBRA permanent data structures in division 2 of the GEANT main store in common /GCBANK/. Creates the user long term division (index IXCONS) (minimum size 2000, maximum size 8*NZEBRA/10). The ZEBRA division IXDIV is reserved for the event data structures and the division IXCONS for the initialisation data structures. Allocates 5200 words of working space. Initialises the link areas and a default run header bank JRUNG [BASE299]. Defines banks format for I/O. GZINIT must be called after GFFGO.

A layout of the dynamic store is shown in fig 3.

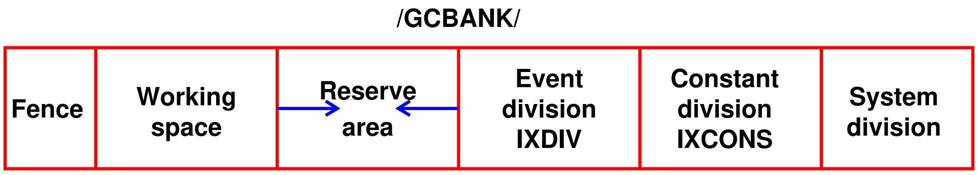


Figure 3: Layout of the dynamic store

CALL **GDINIT**

This routine initialises the GEANT drawing package [DRAW001] and it has to be called before any other graphic routine. GEANT uses the CERN-developed HIGZ [1] graphic library, and this has to be initialised before the call to GDINIT. In the example given in [BASE100] the routines IGINIT and IGMETA are used. Alternatively, the routine HPLINT from HPLOT [7] can be used. This routine calls the appropriate procedures from HIGZ to initialise the underlaying graphics system. At the moment HIGZ can use several flavours of GKS [2, 3, 4] and X11 and it is available on all machines where the CERN Program Library has been installed.

CALL **GPHYSI**

Completes the data structure JMATE, (see [PHYS100]) calculating the cross-section and stopping power tables.

CALL **GBHSTA**

Initialises the standard histograms requested by the user via the data record 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.

CALL **GGCLOS**

This routine has to be called at the end of the definition of the geometry by the user, after thal all volumes have been defined and positioned and all detectors defined. Failure to call this routine will prevent the GEANT system from working correctly. Main tasks of this routine are:

- close the geometry package;
- complete the JVOLUM data structure;
- process the detector definition provided by the user;
- prepare the tables for the tracking speed optimisation requested by the user via the GSORD routine or the OPTI data record.

Origin : R.Brun **Submitted:** 01.06.83
Revision : **Revised:** 26.10.93
Documentation :

Steering routines for event processing

CALL GRUN

Main routine to control a run of events. The following flow chart is only valid for the *batch* execution mode. For interactive applications, see section XINT. A schematic description of the routine is shown in fig. 4.

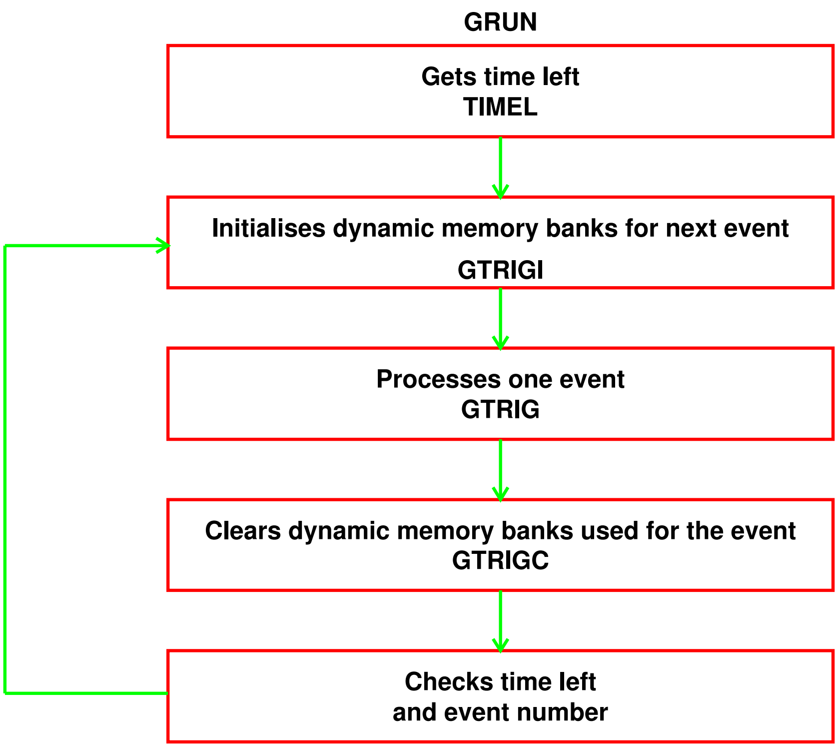


Figure 4: Flow of the GRUN routine.

CALL GTRIGI

Initialisation routine for event processing:

- 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 a default header bank JHEAD for current event [BASE299];
- prints the sequence number, the event number and the random number generator seeds, under control of the flag ITEST (data record DEBU).

CALL GTRIG

Steering routine to process one event (trigger). A schematic description of the routine is shown in fig. 5.
Default routines provided by GEANT are dummy.

CALL GTRIGC

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

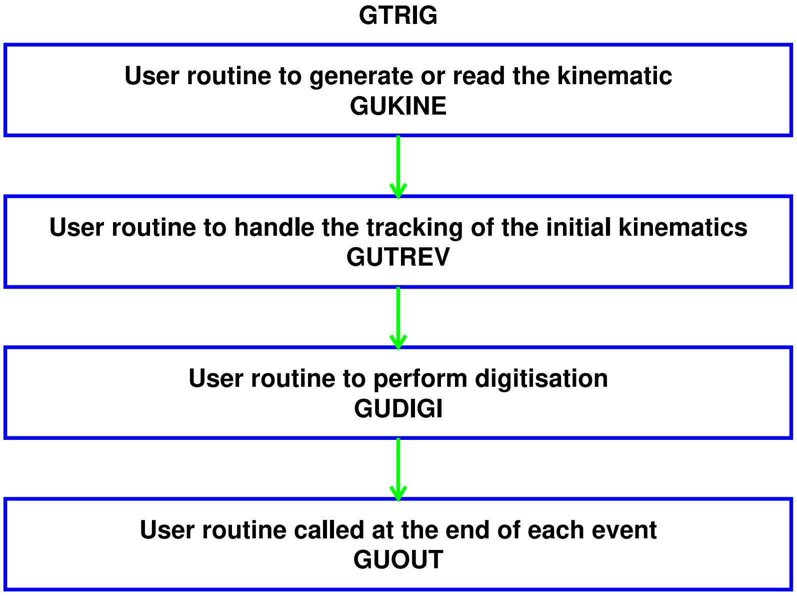


Figure 5: Flow of the GRUN routine.

Origin : M.Maire **Submitted:** 14.12.93
Revision : **Revised:** 14.12.93
Documentation :

Storing and retrieving JRUNG and JHEAD information

CALL **GSRUNG** (NUBUF,UBUF,IADR*)

NUBUF (INTEGER) number of user words;
UBUF (REAL) array of user words;
IADR (INTEGER) position where information is stored in the user bank of the JRUNG structure.

This routine stores the first NUBUF words of array BUF in the user bank attached to the structure JRUNG (see [BASE299]), starting at location IADR+1. On exit IADR is set to IADR+NUBUF, allowing subsequent filling. This allows effectively to *add* information to the current JRUNG bank, whether or not it has already an user buffer.

CALL **GFRUNG** (NWRUNG*,IRUNG*,NUBUF*,UBUF*)

NWRUNG (INTEGER) number of words in JRUNG bank;
IRUNG (REAL) content of JRUNG bank;
NUBUF (INTEGER) number of user words;
UBUF (REAL) array of user words;

This routine retrieves the content of the JRUNG bank and of the user information added, if any.

CALL **GPRUNG**

This routine prints the content of the JRUNG bank and of the user information added, if any.

CALL **GSHEAD** (NUBUF,UBUF,IADR*)

NUBUF (INTEGER) number of user words;
UBUF (REAL) array of user words;
IADR (INTEGER) position where information is stored in the user bank of the JHEAD structure.

This routine stores the first NUBUF words of array BUF in the user bank attached to the structure JHEAD (see [BASE299]), starting at location IADR+1. On exit IADR is set to IADR+NUBUF, allowing subsequent filling. This allows effectively to *add* information to the current JHEAD bank, whether or not it has already an user buffer.

CALL **GFHEAD** (NWHEAD*,IHEAD*,NUBUF*,UBUF*)

NWHEAD (INTEGER) number of words in JHEAD bank;
IHEAD (REAL) content of JHEAD bank;
NUBUF (INTEGER) number of user words;
UBUF (REAL) array of user words;

This routine retrieves the content of the JHEAD bank and of the user information added, if any.

CALL **GPHEAD**

This routine prints the content of the JHEAD bank and of the user information added, if any.

Origin : R.Brun, F.Bruyant
Revision :
Documentation :

Submitted: 01.10.84
Revised: 14.12.93

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`
`+19)`	`NRNDM(1)` First seed for the random number generator at the end of the last event generated
`+20)`	`NRNDM(2)` Second seed for the random number generator at the end of the last event generated
`+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
`+29)`	`IEVENT` event sequence number at the end of the last generated event

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

`LQ(JHEAD-1)`	user link	
`IQ(JHEAD+1)`	`IDRUN`	Run number
`+2)`	`IDEVT`	Event number
`+3)`	`NRNDM(1)`	Value of random number generator first seed at the beginning of the event
`+4)`	`NRNDM(2)`	Value of random number generator second seed at the beginning of the event
`+5 ...10)`		Reserved for user applications

Origin :
Revision :
Documentation : R.Brun

Submitted: 01.10.84
Revised: 08.11.93

Example of user termination routine

```
      SUBROUTINE UGLAST
*
+SEQ,GCLIST
*
*        Call standard GEANT termination routine
      CALL GLAST
*
*        Terminate graphics
      CALL HPLEND
*
*        Close I/O buffers
      IF(NGET .NE. 0 .OR. NSAVE .NE. 0) CALL GCLOSE(0,IER)
*
*        Print histograms
      CALL HISTDO
*
      END
```

CALL GLAST

Standard GEANT termination routine:

- 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 initialised, calls GPSTAT [GEOM700].

Origin : R.Brun, F.Carena **Submitted:** 01.10.84
Revision : **Revised:** 16.12.93
Documentation :

Debugging facilities

The flags IDEBUG, ITEST and ISWIT(1-10) are available to in the common /GCFLAG/ for debug control [ZZZZ010]. The array ISWIT is filled through the data record SWIT. Some flags are used by GHEISHA [PHYS510] and by the routine GDEBUG.

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 record DEBU. If IDEMIN is negative, debug is activated also in the initialisation phase.

The flag ITEST, set by the user via the data record DEBU, is also used by GTRIGI. The sequence number, the event number and the random numbers seeds are printed at the beginning of each event every ITEST from IDEMIN to IDEMAX.

1 Debug of data structures

The contents of the data structures can be dumped by the routine

```
CALL GPRINT (CHNAME,NUMB)
```

CHNAME (CHARACTER*4) name of a top level data structure;
NUMB (INTEGER) number of the 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 are recognised:

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

GPRINT calls selectively the routines:

GPDIGI('*','*')	GPHITS('*','*')	GPKINE(NUMB)	GPMATE(NUMB)
GPVOLU(NUMB)	GPROTM(NUMB)	GPSETS('*','*')	GPTMED(NUMB)
GPPART(NUMB)	GPVERT(NUMB)	GPJXYZ(NUMB)	

These routines can be called directly by the user. In case of SETS, HITS and DIGI the content of all detectors of all sets will be printed, so NUMB is irrelevant.

2 Debug of events

The development of an event can be followed via the routine:

```
CALL GDEBUG
```

which operates under the control of the ISWIT array. It is the user responsibility to call this routine from GUSTEP. If the DEBUG flag is active, the routine will perform as follows:

ISWIT(1)

2 the content of the temporary stack for secondaries in the common /GCKING/ is printed;

ISWIT(2)

1 the current point of the track is stored in the JDXYZ bank via the routine GSXYZ;
2 the current information on the track is printed via the routine GPCXYZ;
3 the current step is drawn via the routine GDCXYZ;
4 the current point of the track is stored in the JDXYZ bank via the routine GSXYZ. When the particle stops the track is drawn via the routine GDTRAK and the space occupied by the track in the structure JDXYZ released;

ISWIT(3)

1 the current point of the track is stored in the JDXYZ bank via the routine GSXYZ.

Origin : R.Brun
Revision :
Documentation :

Submitted: 08.08.87
Revised: 08.11.93

Utility Routines

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

CHNAME (CHARACTER*4) variable containing the name to be searched for in IVECT;
IVECT (INTEGER) array containing the ASCII code of the names among which NAME is searched. The names are stored 4 characters per word;
N (INTEGER) number of items in IVECT;
ILOOK (INTEGER) position in IVECT where NAME has been found, 0 if not found.

This routine is very useful when searching for a string stored in a ZEBRA bank. For instance to find the position of the 'CRYS' volume in the volume bank, the following piece of code could be written:

```
+SEQ,GCBANK.
+SEQ,GCNUM.
      .
      .
      .
      CALL GLOOK('CRYS',IQ(JVOLUM+1),NVOLUM,IVO)
      IF(IVO.GT.0) THEN
         JVO = LQ(JVOLUM-IVO)
      ELSE
         JVO = 0
      ENDIF
```

JVO, if different from 0, is the pointer to the data bank containing the information relative to the volume 'CRYS'.

CALL **GEVKEV** (EGEV,ENERU*,CHUNIT*)

EGEV (REAL) input, energy in GeV;
ENERU (REAL) output, energy in the new unit;
CHUNIT (CHARACTER*4) unit in which the energy has been converted.

This subroutine converts the input energy in GeV to a unit in which $1 \leq E \leq 999$. CHUNIT contains the new unit. The following piece of code illustrates the use of GEVKEV:

```
+SEQ,GCTRAK.
      CHARACTER*4 CHUNIT
      .
      .
      .
      CALL GEVKEV(DESTEP, DE, CHUNIT)
      WRITE(6,10000) DE, CHUNIT
10000 FORMAT(' The energy loss in this step is ',F7.2,A)
```

Origin : R.Brun, F.Carminati
Revision :
Documentation :

Submitted: 27.07.93
Revised: 14.12.93

The random number generator

CALL **GRNDM** (VEC*,LEN)

VEC (REAL) vector containing the generated random number;
LEN (INTEGER) number of random numbers to generate.

GRNDM generates a sequence of uniformly distributed random numbers in the interval (0,1). The numbers are returned in a vector. The code is a copy of the CERN Program Library routine RANECU [16, 17] (entry V114).

Several independent sequences can be defined and used. Each sequence **must** be initialised by the user, otherwise the result is unpredictable. Two integer seeds are used to initialise a sequence. Not all pairs of integers define a good random sequence or one which is independent from others. Sections of the same random sequence can be defined as independent sequences. The period of the generator is $2^{60} \approx 10^{18}$. A generation has been performed in order to provide the seeds to start any of the generated sections. There are 215 possible seed pairs and they are all 10^9 numbers apart. Thus a sequence started from one of the seed pairs, after 10^9 numbers will start generating the next one.

CALL **GRNDMQ** (ISEED1,ISEED2,ISEQ,CHOPT)

ISEED1 (INTEGER) first seed of the sequence;
ISEED2 (INTEGER) second seed of the sequence;
ISEQ (INTEGER) number of the independent sequence of random numbers referred to by this call. If ISEQ$\leq$0, then the last valid sequence used will be addressed either for a save or a store. In case the option 'G' is specified, on output the variable will contain the sequence actually used;
CHOPT (CHARACTER*(*)) the action to be taken:

- ' ' if 1$\leq$ISEQ$\leq$215, sequence number ISEQ will be initialised with the seeds of the pre-computed independent sequence ISEQ.
  If ISEQ$\leq$0 or ISEQ>215, sequence number 1 will be initialised with the default seeds. ISEED1 and ISEED2 are ignored;
- 'G' get the present status of the generator: the two integer seeds ISEED1 and ISEED2 will be returned for sequence ISEQ;
- 'S' set the status of the generator. The two integer seeds ISEED1 and ISEED2 will be used to restart the generator for sequence ISEQ.
- 'SH' the same action as for 'S' and store the two integer seeds ISEED1 and ISEED2 in the event header bank.
- 'Q' Get the pre-generated seeds for ISEQ (1$\leq$ISEQ$\leq$215). There are 215 pre-generated sequences each one will generate 10^9 numbers before reproducing the following one.

Initialises the random number generator.

VALUE = **GARNDM** (DUMMY)

DUMMY (REAL) dummy parameter, ignored;

Returns a random number r distributed as e^{-x}. $r = -\ln(\eta)$ where η is a random number extracted by GRNDM.

CALL **GPOISS** (AMVEC,NPVEC*,LEN)

AMVEC (REAL) array of length LEN containing the average values of the Poisson distributions requested;
NPVEC (INTEGER) array of length LEN containing the random numbers: NPVEC(I) is a random number with a Poisson distribution of average AMVEC(I);
LEN (INTEGER) number of random numbers requested.

This routine extracts integer random numbers according to the Poisson distribution. Given a Poisson distribution of average $\lambda \leq 16$ and r uniformly distributed between 0 and 1, the method used is the following:

1. let $P = \exp(-\lambda)$, $N = 0$ and $S = P$;
2. if $r \leq S$ accept N;
3. let $N = N + 1$, $P = P\lambda/N$, $S = S + P$ and go back to 2;

If $\lambda > 16$, a gaussian distribution with average λ and $\sigma = \sqrt{\lambda}$ is generated.

BASE Bibliography

[1] H.C.Fesefeldt. Simulation of hadronic showers, physics and applications. Technical Report PITHA 85-02, III Physikalisches Institut, RWTH Aachen Physikzentrum, 5100 Aachen, Germany, September 1985.

[2] P.A.Aarnio et al. Fluka user's guide. Technical Report TIS-RP-190, CERN, 1987, 1990.

[3] A.Fassò A.Ferrari J.Ranft P.R.Sala G.R.Stevenson and J.M.Zazula. FLUKA92. In *Proceedings of the Workshop on Simulating Accelerator Radiation Environments*, Santa Fe, USA, 11-15 January 1993.

[4] A.Fassò A.Ferrari J.Ranft P.R.Sala G.R.Stevenson and J.M.Zazula. A Comparison of FLUKA Simulations with measurements of Fluence and Dose in Calorimeter Structures. *Nuclear Instruments & Methods A*, 332:459, 1993.

[5] C.Birattari E.DePonti A.Esposito A.Ferrari M.Pelliccioni and M.Silari. Measurements and characterization of high energy neutron fields. *approved for pubblication in Nuclear Instruments & Methods A.*

[6] P.A.Aarnio, A.Fassò, A.Ferrari, J.-H.Moehring, J.Ranft, P.R.Sala, G.R.Stevenson and J.M.Zazula. FLUKA: hadronic benchmarks and applications. In *MC93 Int. Conf. on Monte-Carlo Simulation in High-Energy and Nuclear Physics*, Tallahassee, Florida, 22-26 February 1993. Proceedings in press.

[7] R.Brun et al. *FFREAD User Guide and Reference Manual*, dd/us/71 edition, February 1987.

[8] H.J. Klein and J. Zoll. *PATCHY Reference Manual,* Program Library L400. CERN, 1988.

[9] M. Brun, R. Brun, and F. Rademakers. *CMZ - A Source Code Management System*. CodeME S.A.R.L., 1991.

[10] R.Brun, M.Goossens, and J.Zoll. *ZEBRA Users Guide,* Program Library Q100. CERN, 1991.

[11] R.Bock et al. *HIGZ Users Guide,* Program Library Q120. CERN, 1991.

[12] R.Brun and H.Renshall. *HPLOT users guide,* Program Library Y251. CERN, 1990.

[13] ISO. The graphical kernel system (gks). Technical Report ISO 7942, ISO Geneva, 1985.

[14] ISO. The graphical kernel system for three dimensions (gks-3d). Technical Report ISO 8805, ISO Geneva, 1987.

[15] D.R.Myers. *GKS/GKS-3D Primer,* DD/US/110. CERN, 1987.

[16] P.L'Ecuyer. Efficient and portable random number generators. *Comm. ACM*, 31:742, 1988.

[17] F.James. A review of pseudorandom number generators. *Computer Phys. Comm.*, 60:329–344, 1990.

Origin :
Revision :
Documentation : F.Bruyant

Submitted: 01.10.84
Revised: 17.12.93

Introduction to the section CONS

1 Introduction

The setup where the particles are transported is represented by a structure of geometrical volumes. Each volume is filled with matter (which can have the properties of the vacuum in case of it contains no matter).

The matter composing the volumes is described via two sets of attributes. The first set is relative to the nature of the *material* composing the volume, and contains information such as the atomic number, the atomic weight, the density and so on (see the description of the routine GSMATE in this section for more information).

The second set of attributes is relevant to the process of particle transport and they define a so-called *tracking medium*. These are parameters such as the material composing the volume, the magnetic field, the required tracking precision, the maximum energy that can be lost in one step and so on (see the description of the routine GSTMED in this section for more information).

Each tracking medium refers to a material via a *material number* which is assigned by the user. Different tracking media can, with some limitation, refer to the same material.

Each volume is filled by a tracking medium identified by a *medium number*. Different volumes may have the same medium number (see [GEOM]).

The transport of particles through a setup ([TRAK]) requires access to data which describe:

- the geometry of the setup;
- the *material* and *tracking medium* parameters;
- the particle properties.

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

Important note: many entities in GEANT are user-defined via a subroutine call. One of the arguments of this subroutine is a integer number which identifies the entity. Examples are materials, tracking media, particles and so on. It can be tempting, for booking purposes, to use very large numbers. For instance, in a large detector, the number of all the materials in the hadronic calorimeter could be $1001 \leq I \leq 2000$. Even if these conventions are very handy, they can introduce a performance penalty.

For reasons of speed, the number provided by the user is used directly as the number of the link in the ZEBRA data structure indicating where to store the pointer to the bank containing the information on the entity. ZEBRA pointers are contiguous. Defining an object with a user number of 1000 forces ZEBRA to allow space for 1000 links. This entails a loss of space (999 words), but much worse than that, induces ZEBRA to believe that there are in fact 999 more banks. At every operation which causes a *relocation* of banks, ZEBRA will check all the possible links, which can be very time consuming.

So values of user-defined entities must be kept as small as possible and contiguous. In large applications one could write a routine which returns the next free number to be allocated, which can then be stored in a variable and always referenced symbolically, freeing the user from the need to define ranges. As an example we give here a function performing this operation for the material number:

```
      FUNCTION NEXTMA()
+SEQ, GCBANK
      IF(JMATE.LE.0) THEN
```

```
         NEXTMA = 1
      ELSE
         DO 10 IMAT = 1, IQ(JMATE-2)
            IF(LQ(JMATE-IMAT).EQ.0) GOTO 20
 10      CONTINUE
 20      NEXTMA = IMAT
      ENDIF
      END
```

2 Materials

The material constants are stored in the data structure JMATE via the routine GSMATE and can be retrieved via the routine GFMATE and printed via the routine GPMATE. The routine GMATE calls repeatedly the routine GSMATE to define a standard set of materials, but its use is not mandatory. Mixtures of basic materials, compounds or molecules with atoms from different basic materials, may also be defined and their characteristics are stored in the structure JMATE, through calls to the routine GSMIXT. Mixtures of compounds are allowed.

Quantities such as energy loss and cross-section tables (see [PHYS]), computed during the initialisation of the program are stored in the structure JMATE. These can be accessed through the routine GFTMAT and plotted or printed through the routines GPLMAT and GPRMAT respectively.

3 Tracking media

The tracking medium parameters are stored in the data structure JTMED by 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 correctness of the transport process and thus the reliability of the results obtained with GEANT depend critically on the values of the *tracking medium parameters*. To help the user, most of the parameters are recalculated by default by GEANT irrespective of the value assigned by the user. Advanced users can control the value of the parameters thus bypassing the automatic computation. See the description of the routine GSTMED for more information.

The tracking thresholds, and other parameters and flags that control the physics processes, defined in GINIT and possibly modified through the relevant data records, are also stored in the structure JTMED. These parameters can be redefined for each tracking medium via the routine GSTPAR.

4 Particles

The parameters of the particles transported by GEANT are stored in the data structure JPART via the routine GSPART. The decay properties of particles are defined via the routine GSDK. The particle constants can be accessed with the routine GFPART and printed with the routine GPPART.

The routine GPART defines the standard table of particles, branching ratios and decay modes. This routine needs to be called for the GEANT system to work properly. GPART calls the routine GSPART for the particle properties and GSDK for the decays for each particle in the standard GEANT table. The user may call GSPART and GSDK to extend or modify the table defined by GPART.

Origin : R.Brun, G.N.Patrick
Revision :
Documentation :

Submitted: 06.06.83
Revised: 01.11.93

Material definition

`CALL` **GMATE**

Stores the following standard material constants in the data structure [7] `JMATE`.

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

Note: this table provides some material definition with standard properties. Advanced user will want to define their own table of materials, which can be done with the routine `GSMATE`, described below.

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

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

`IMATE` (`INTEGER`) material number;
`CHNAMA` (`CHARACTER*20`) material name;
`A` (`REAL`) atomic weight;
`Z` (`REAL`) atomic number;
`DENS` (`REAL`) density in g cm^{-3};
`RADL` (`REAL`) radiation length in cm;
`ABSL` (`REAL`) absorption length in cm. This parameter is ignored by `GEANT`, but it has been kept in the calling sequence for backward compatibility;
`UBUF` (`REAL`) array of `NWBUF` additional user parameters;
`NWBUF` (`INTEGER`) number of user words in `UBUF`.

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

Retrieves the constants of material `IMATE` from the data structure `JMATE`. The parameters are the same as for `GSMATE`.

`CALL` **GPMATE** `(IMATE)`

Prints the material constants:

`IMATE` (`INTEGER`) material number. If `IMATE=0` then all materials will be printed. If `IMATE<0` the material number `-IMATE` will be printed without the header describing the value of the various fields.

Origin : R.Brun, M.Maire **Submitted:** 23.04.86
Revision : **Revised:** 03.11.93
Documentation :

Retrieve material cross-sections and stopping power

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

Retrieve and **interpolate** the dE/dx and cross-sections tabulated in material banks corresponding to a given array of kinetic energy bins.

IMATE	(INTEGER) material number;
IPART	(INTEGER) particle number;
CHMECA	(CHARACTER*4) name of the mechanism bank to be retrieved;
KDIM	(INTEGER) dimension of the arrays TKIN and VALUE;
TKIN	(REAL) array of kinetic energies in GeV;
VALUE	(REAL) array of energy losses in MeV cm^{-1}, or macroscopic cross sections in cm^{-1};
PCUT	(REAL) array containing the 5 energy thresholds for the material in GeV (CUTGAM, CUTELE, CUTNEU, CUTHAD, CUTMUO);
IXST	(INTEGER) return code: 0 the table was not filled: 1 normal return; values are returned into the VALUE array;

The mechanisms have the following conventional name and a code number (IMECA):

HADF (1)	total hadronic interaction cross-section according to FLUKA;
INEF (2)	hadronic inelastic cross-section according to FLUKA;
ELAF (3)	hadronic elastic cross-section according to FLUKA;
NULL (4)	unused;
NULL (5)	unused;
HADG (6)	total hadronic interaction cross-section according to GHEISHA;
INEG (7)	hadronic inelastic cross-section according to GHEISHA;
ELAG (8)	hadronic elastic cross-section according to GHEISHA;
FISG (9)	nuclear fission cross-section according to GHEISHA;
CAPG (10)	neutron capture cross-section according to GHEISHA;
LOSS (11)	stopping power;
PHOT (12)	photoelectric cross-section;
ANNI (13)	positron annihilation cross-section;
COMP (14)	Compton effect cross-section;
BREM (15)	bremsstrahlung cross-section;
PAIR (16)	photon and muon direct- e^-e^+pair cross-section;
DRAY (17)	δ-rays cross-section;
PFIS (18)	photo-fission cross-section;
RAYL (19)	Rayleigh scattering cross-section;
MUNU (20)	muon-nuclear interaction cross-section;
RANG (21)	range in cm;
STEP (22)	maximum step in cm.

Note: Common /GCMULO/ contains an array ELOW(200) (see [CONS199]) with NEK1 kinetic energy values ranging from EKMIN to EKMAX. GPHYSI initialises by default the first 91 locations of ELOW with values of energy from 10^{-5} GeV (10 keV) to 10^4 GeV (10 TeV) equally spaced on logarithmic scale. This can be controlled by the user via the data record or the interactive command `ERAN` ([BASE040]). `ELOW` can be used as the input argument `TKIN`, e.g.:

```
CALL GFTMAT (10, 1,'PHOT',NEK1, ELOW ,VALUE, PCUT, IXST)
```

will return in array `VALUE` the photoelectric cross-section in material number 10.

`CALL` **GPLMAT** `(IMATE,IPART,CHMECA,KDIM,TKIN,IDM)`

Plot and **interpolate** the dE/dx and cross-sections tabulated in material banks corresponding to a given array of kinetic energy bins. The mechanisms are the same as the previous routine. If the mechanism name is `'ALL'`, then all the mechanisms will be plotted.

The first five parameters have the same meaning as in the previous routine and:

`IDM` (`INTEGER`) treatment of the created histogram(s):

- > 0 fill, print, keep histogram(s);
- $= 0$ fill, print, delete histogram(s);
- < 0 fill, noprint, keep histogram(s).

The result is not returned to the user as is the case in the previous routine but it is put in an `HBOOK` histogram. The `HBOOK` package needs to be initialised by the user via the routine `HLIMIT`, see [BASE100] for more information. The histogram identifier is calculated as: `10000*IMATE+100*IPART+IMECA` where `IMECA` is the mechanism code listed above.

`CALL` **GPRMAT** `(IMATE,IPART,CHMECA,KDIM,TKIN)`

Print and **interpolate** the dE/dx and cross-sections tabulated in material banks corresponding to a given array of kinetic energy bins. The mechanisms are the same as the previous routine. If the mechanism name is `'ALL'`, then all the cross sections will be printed.

The result is printed in the form of a table.

The five parameters have the same meaning as in the previous routine.

Origin : R.Brun, M.Maire, J.Allison
Revision :
Documentation : M.Maire

Submitted: 01.06.83
Revised: 08.12.93

Mixtures and Compounds

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

Defines mixture or compound `IMATE` as composed by `NLMAT` materials defined via the arrays `A`, `Z` and `WMAT`. Mixtures of compounds can also be defined.

`IMATE`	(`INTEGER`) user material (mixture) number;
`NAMATE`	(`CHARACTER*20`) mixture name;
`A`	(`REAL`) array of atomic weights;
`Z`	(`REAL`) array of atomic numbers;
`DENS`	(`REAL`) density in g cm^{-3};
`NLMAT`	(`INTEGER`) number of elements in the mixture;
	> 0 `WMAT` contains the proportion by weights of each material in the mixture;
	< 0 `WMAT` contains the proportion by number of atoms of each kind, the content of `WMAT` in output is changed to contain the relative weights;

Method

For a compound (`NLMAT` < 0), the *molecular* weight and charge are:

$$A_{mol} = \sum_i n_i A_i \qquad Z_{mol} = \sum_i n_i Z_i$$

where n_i = `WMAT(I)` is the number of atoms of the ith component of the molecule. In this case the proportion by weight is:

$$p_i = \frac{n_i A_i}{A_{mol}}$$

where p_i = `WMAT(I)` in output. From the relative weights, `GSMIXT` works out the effective atomic weight and atomic number according to the following formulas:

$$A_{eff} = \sum_i p_i A_i \qquad Z_{eff} = \sum_i p_i Z_i$$

which are stored in the `JMATE` data structure [CONS199] together with the radiation length.

The radiation length is computed according to the EGS manual [12], formula 268 (37), for an element:

$$\frac{1}{\rho X_0} = 4\alpha r_0^2 N_{Av} \frac{1}{A} Z(Z + \xi(Z)) \left[\ln \frac{183}{Z^{1/3}} - F_c(Z) \right]$$

where

X_0	radiation length (in cm);
ρ	density (in g cm^{-3});
α	fine structure constant;
r	classical electron radius (in cm);
N_{Av}	Avogadro's number;

A	atomic weight;
Z	atomic number;
$F_c(Z)$	Coulomb correction function.

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

$$\xi(Z) = \frac{\ln \frac{1440}{Z^{2/3}}}{\ln \frac{183}{Z^{1/3}} - F_c(Z)}$$

for a compound or mixture:

$$\frac{1}{\rho X_0} = \sum_i \frac{p_i}{\rho_i X_{0i}}$$

where p_i is the proportion by weight of the ith element. For more information on the organisation of the data in memory see `[CONS199]`.

The effective absorption length λ is defined as the interaction length of a 5GeV pion in the material:

$$\frac{1}{\lambda} = \sum_i \frac{p_i}{\lambda_i} \qquad \frac{1}{\lambda_i} = \frac{\sigma_i^\pi(5GeV)N_{Av}}{A_i} \quad \mathrm{g}^{-1}\ \mathrm{cm}^2$$

Once this value has been determined, an effective hadronic atomic weight (A_{h-eff}) is calculated by the routine `GHMIX` [3] and stored in the data structure.

Origin : R.Brun
Revision :
Documentation : R.Brun, M.Maire

Submitted: 01.06.83
Revised: 23.03.94

The Material data structure JMATE

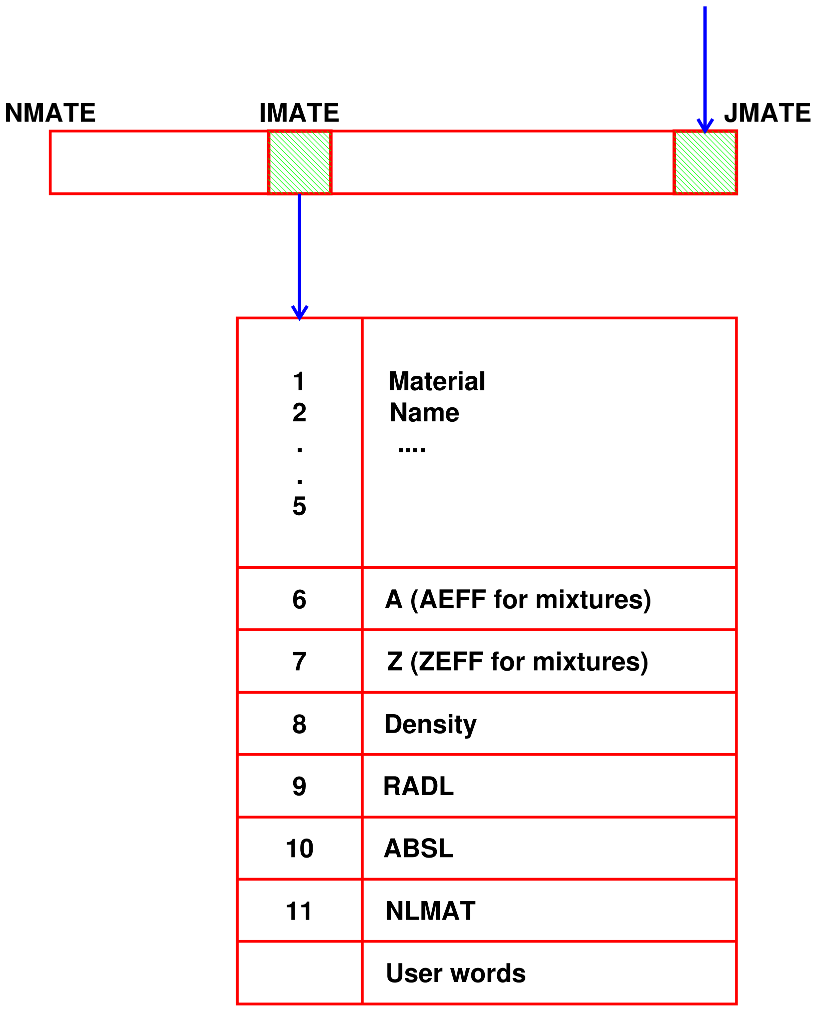


Figure 6: `JMA = LQ(JMATE-IMATE)` pointer to material `IMATE`

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.

Note: The energy losses are stored in $GeV\ cm^{-1}$. The inverse of the macroscopic cross-section (i.e. the mean free path, in cm), is stored instead of the cross-section, to speed up the calculation of the distance to the next interaction point.

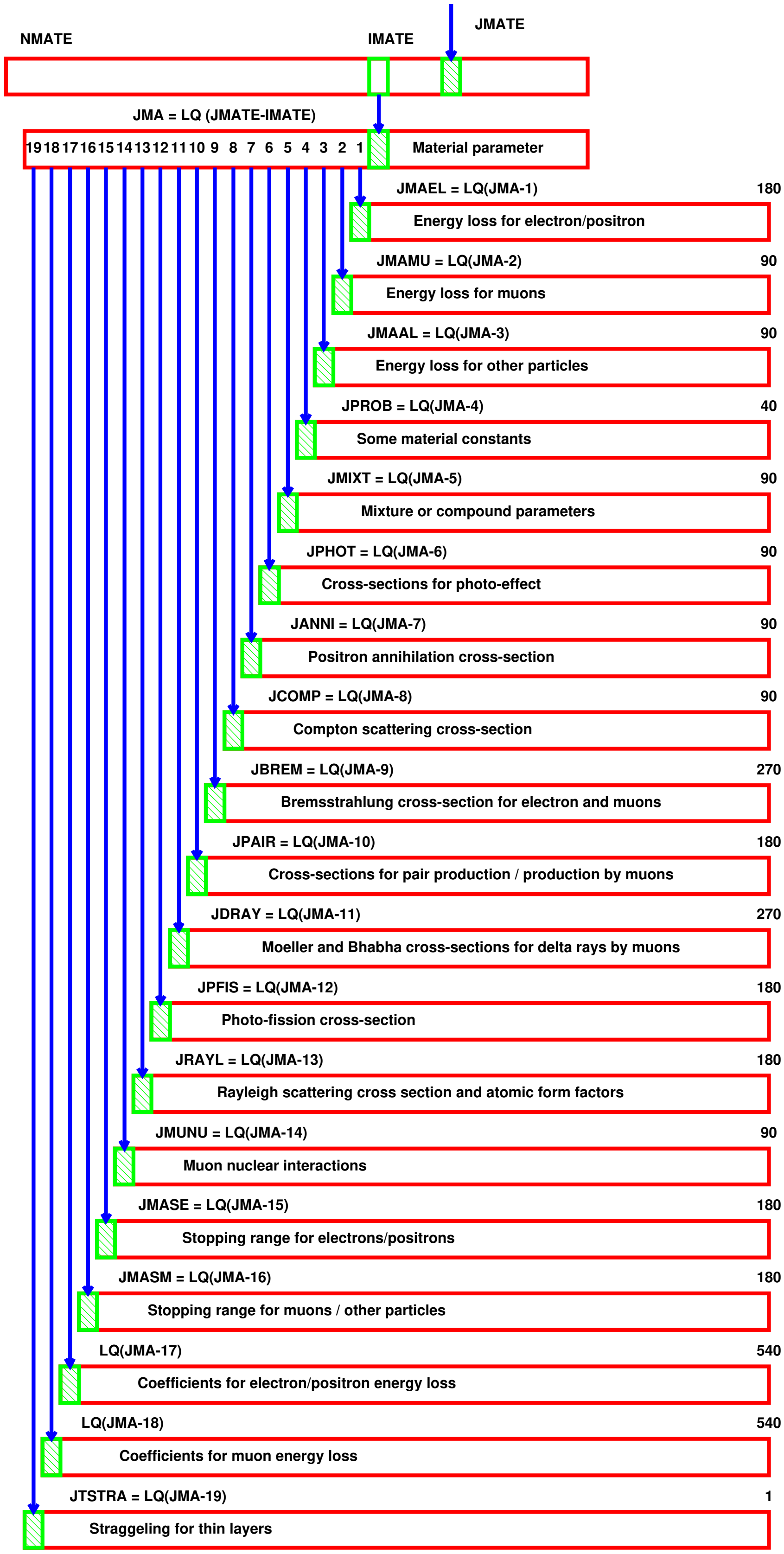


Figure 7: Material data structure

1 Energy binning

IDECAD	Bin number	Energy range IEKBIN
1	$1 \rightarrow 10$	10 KeV $\rightarrow$ 79.43 KeV
2	$11 \rightarrow 20$	100 KeV $\rightarrow$ 794.3 KeV
3	$21 \rightarrow 30$	1 MeV $\rightarrow$ 7.943 MeV
4	$31 \rightarrow 40$	10 MeV $\rightarrow$ 79.43 MeV
5	$41 \rightarrow 50$	100 MeV $\rightarrow$ 794.3 MeV
6	$51 \rightarrow 60$	1 GeV $\rightarrow$ 7.943 GeV
7	$61 \rightarrow 70$	10 GeV $\rightarrow$ 79.43 GeV
8	$71 \rightarrow 80$	100 GeV $\rightarrow$ 794.3 GeV
9	$81 \rightarrow 90$	1 TeV $\rightarrow$ 7.943 TeV

The values of the bins are kept in the array ELOW(90) in /GCMULO/ :

$$\mathrm{ELOW}(i) = 10^{\frac{i-1}{10}-5}[\mathrm{GeV}]$$

2 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].

3 Energy loss for muons

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

4 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].

5 Some material constants

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

See routine GPROBI.

6 Mixture and compound parameters

Words 1 to 4*NLM where NLM is the number of mixture or compound components [CONS110].

7 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].

8 Positron annihilation

[PHYS 350].

9 Compton scattering

[PHYS 220].

10 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].

11 Pair production by gammas and muons

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

12 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 $e^- \rightarrow e^-$ [PHYS 430].

13 Photo-fission

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

Origin : R.Brun **Submitted:** 01.06.83
Revision : F.Carminati, M.Maire **Revised:** 08.12.93
Documentation :

Tracking medium parameters

Description of the routines

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

This routine associates a set of tracking parameters to a material, defining a so-called *tracking medium*. Every GEANT volume must be associated to an existing tracking medium. The routine stores the parameters of the tracking medium ITMED in the data structure JTMED.

- ITMED (INTEGER) tracking medium number;
- NATMED (CHARACTER*20) tracking medium name;
- NMAT (INTEGER) material number corresponding to ITMED;
- ISVOL (INTEGER) *sensitivity* flag (see later):
  - $\leq$0 not a sensitive volume;
  - $>$0 sensitive volume;
- IFIELD (INTEGER) magnetic field flag:
  - =0 no magnetic field;
  - =1 strongly inhomogeneous magnetic field (returned by the user function GUFLD): tracking performed with the Runge-Kutta method by the routine GRKUTA;
  - =2 inhomogeneous magnetic field (returned by the user function GUFLD), tracking along a helix performed by the routine GHELIX;
  - =3 uniform magnetic field along the z axis of strength FIELDM, tracking performed along a helix by the routine GHELX3;
- FIELDM (REAL) maximum field value (in Kilogauss);
- TMAXFD (REAL) maximum angular deviation due to the magnetic field permitted in one step (in degrees);
- STEMAX (REAL) maximum step permitted (cm);
- DEEMAX (REAL) maximum fractional energy loss in one step ($0 <$ DEEMAX ≤ 1);
- EPSIL (REAL) boundary crossing precision (cm);
- STMIN (REAL) minimum value for the maximum step imposed by energy loss, multiple scattering, Čerenkov or magnetic field effects (cm);
- UBUF (REAL) array of NWBUF additional user parameters;
- NWBUF (INTEGER) number of additional user parameters.

Note: the tracking medium number can in principle be any positive integer from 1 to 65,535. However this number is used directly by GEANT as the number of the pointer to the data structure where the information is stored. When a pointer is defined, all pointers from 1 to the one allocated are created, if not already existing. Every time data structures are moved in memory, all the links are potentially scanned for update. This can be time consuming and it can seriously affect performances. So a continuous range of numbers should be used for tracking media.

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

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

```
CALL GPTMED (ITMED)
```

Prints the tracking medium parameters for the given medium.

`ITMED` (`INTEGER`) tracking medium to be printed, all tracking media if `ITMED`=0.

Automatic calculation of the tracking parameters

The results of the simulation depend critically on the choice of the tracking medium parameters. To make of `GEANT` a reliable and consistent predictive tool, the calculation of these parameters has been automated. In a normal GEANT run, the variable `IGAUTO` in common `/GCTRAK/` is set to 1. In this case the following parameters are calculated by the program:

$$\mathtt{TMAXFD} = \begin{cases} 20^\circ & \text{if } \mathtt{TMAXFD} \le 0 \text{ or } \mathtt{TMAXFD} > 20 \\ \textit{input value} & \text{if } 0 < \mathtt{TMAXFD} \le 20 \end{cases}$$

$$\mathtt{STEMAX} = \mathtt{BIG}(=10^{10})$$

$$\mathtt{DEEMAX} = \begin{cases} 0.25 & \text{if } \mathtt{ISVOL} = 0 \text{ and } X_0 \le 2cm\text{, where } X_0 \text{ is the radiation length of the material} \\ 0.25 - \dfrac{0.2}{\sqrt{X_0}} & \text{otherwise} \end{cases}$$

$$\mathtt{STMIN} = \begin{cases} \dfrac{2R}{\sqrt{X_0}} & \text{if } \mathtt{ISVOL} = 0 \\ \dfrac{5R}{\sqrt{X_0}} & \text{if } \mathtt{ISVOL} \ne 0 \end{cases}$$

where R is the range in cm of an electron of energy `CUTELE`+200keV.

If the `IGAUTO` variable is set to 0 via the `AUTO` data record, then value given by the user for the above parameters is accepted as the true parameter value if > 0, while automatic calculation still takes place in case the input value is negative.

Users are strongly recommended to begin their simulation with the parameters as calculated by `GEANT`. Users who want to modify any of the above parameters must be sure they understand their function in the program and the implications of a change.

The `EPSIL` parameter represents the absolute precision with which the tracking is performed. It has a double meaning. When tracking in magnetic field, `EPSIL` is the minimum distance for which boundaries are checked. A particle nearer than `EPSIL` to the boundary is considered as exiting the current volume. If the end point of the step of a particle in magnetic field is distant less than `EPSIL` along each axis from what would be the end point in absence of magnetic field, then no boundary checking is performed.

In all cases, when a particle is going to reach the boundary of the current volume with the current step, the step length is increased by a quantity (`PREC`, common `/GCTMED/`) which is set to the minimum between $0.1 \times EPSIL$ and 10 micron at the beginning of the tracking. This quantity is recalculated at every step according to the formula:

$$\mathtt{PREC} = \max\left[\min\left(\frac{\mathtt{EPSIL}}{10}, 10\mu\right), \max\left[|x|, |y|, |z|, S\right) \times \epsilon\right] \tag{1}$$

Where x, y, z are the current coordinates of the particle, S is the length of the track, and ϵ is the assumed machine precision. ϵ (called `EPSMAC` in the program) is initially set to 10^{-6} for 32 bits machines and 10^{-11} for 64 bits machines. During the tracking, every fifth time that a particle tries unsuccessfully to exit from the same volume, the machine precision is multiplied by the number of trials. This accounts for additional losses of precision due to transformation of coordinates and region of the floating point range where the real machine precision is different from the above (this happens in particular on IBM mainframes with 370 floating point number representation).

Origin : R.Brun,F.Bruyant **Submitted:** 11.02.86
Revision : **Revised:** 08.12.93
Documentation : same

Special Tracking Parameters

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

`ITMED` (`INTEGER`) tracking medium number;
`CHPAR` (`CHARACTER*4`) character string, name of the variable to be modified;
`PARVAL` (`REAL`) new value (must be given as a floating point).

Default parameters	Parameter name	Default value
Q(JTMN +1) = CUTGAM	CUTGAM	0.001Gev
Q(JTMN +2) = CUTELE	CUTELE	0.001Gev
Q(JTMN +3) = CUTNEU	CUTNEU	0.01Gev
Q(JTMN +4) = CUTHAD	CUTHAD	0.01Gev
Q(JTMN +5) = CUTMUO	CUTMUO	0.01Gev
Q(JTMN +6) = BCUTE	BCUTE	CUTGAM
Q(JTMN +7) = BCUTM	BCUTM	CUTGAM
Q(JTMN +8) = DCUTE	DCUTE	10.Tev
Q(JTMN +9) = DCUTM	DCUTM	10.Tev
Q(JTMN+10) = PPCUTM	PPCUTM	0.01Gev
Q(JTMN+11) = IPAIR	PAIR	1
Q(JTMN+12) = ICOMP	COMP	1
Q(JTMN+13) = IPHOT	PHOT	1
Q(JTMN+14) = IPFIS	PFIS	0
Q(JTMN+15) = IDRAY	DRAY	2
Q(JTMN+16) = IANNI	ANNI	1
Q(JTMN+17) = IBREM	BREM	1
Q(JTMN+18) = IHADR	HADR	1
Q(JTMN+19) = IMUNU	MUNU	0
Q(JTMN+20) = IDCAY	DCAY	1
Q(JTMN+21) = ILOSS	LOSS	2
Q(JTMN+22) = IMULS	MULS	1
Q(JTMN+26) = GHEISHA	GHCOR1	see [PHYS700]
Q(JTMN+27) = MODEL	BIRK1	see [PHYS337]
Q(JTMN+28) = RKB	BIRK2	see [PHYS337]
Q(JTMN+29) = C	BIRK3	see [PHYS337]
Q(JTMN+31) = ILABS	LABS	see [PHYS260]
Q(JTMN+32) = ISYNC	SYNC	see [PHYS360]
Q(JTMN+33) = ISTRA	STRA	see [PHYS333]

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. For example to change CUTGAM to 0.0001 for tracking medium ITMED:

```
CALL GSTPAR(ITMED,'CUTGAM',0.0001)
```

For more information on default values for IDRAY and ILOSS see [BASE040].

The default tracking medium parameters are stored in the bank whose pointer is stored in the JTMED variable. In this case in the above table JTMN = JTMED. In the case one or more of the above parameters has been modified for a specific tracking medium, the whole parameter set is stored in the next bank of the given trackin medium. In this case JTMN = LQ(JTM) in the above table, where JTM is the pointer to the bank of the tracking medium, i.e. JTM = LQ(JTMED-ITMED).

Note

At tracking time the parameters above are copied from JTMN into the common blocks /GCCUTS/ and /GCPHYS/. GSTPAR must be called before GPHYSI.

Origin : R.Brun, G.N.Patrick
Revision :
Documentation :

Submitted: 01.06.83
Revised: 13.05.92

Particle definition

`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	0	stable (10^{15})
Positron	2	0.00051099906	1	stable
Electron	3	0.00051099906	-1	stable
Neutrino	4	0.0	0	stable
Muon +	5	0.105658389	1	2.19703×10^{-6}
Muon -	6	0.105658389	-1	2.19703×10^{-6}
Pion 0	7	0.1349764	0	8.4×10^{-17}
Pion +	8	0.1395700	1	2.603×10^{-8}
Pion -	9	0.1395700	-1	2.603×10^{-8}
Kaon 0 long	10	0.497672	0	5.17×10^{-8}
Kaon +	11	0.493677	1	1.237×10^{-8}
Kaon -	12	0.493677	-1	1.237×10^{-8}
Neutron	13	0.93956563	0	887.0
Proton	14	0.93827231	1	stable
Antiproton	15	0.93827231	-1	stable
Kaon 0 short	16	0.497672	0	8.926×10^{-11}
Eta	17	0.54745	0	5.485×10^{-19}
Lambda	18	1.115684	0	2.632×10^{-10}
Sigma +	19	1.18937	1	7.99×10^{-11}
Sigma 0	20	1.19255	0	7.40×10^{-20}
Sigma -	21	1.197436	-1	1.479×10^{-10}
Xi 0	22	1.3149	0	2.90×10^{-10}
Xi -	23	1.32132	-1	1.639×10^{-10}
Omega	24	1.67245	-1	8.22×10^{-11}
Antineutron	25	0.93956563	0	887.0
Antilambda	26	1.115684	0	2.632×10^{-10}
Antisigma -	27	1.18937	-1	7.99×10^{-11}
Antisigma 0	28	1.19255	0	7.40×10^{-20}
Antisigma +	29	1.197436	1	1.479×10^{-10}
Antixi 0	30	1.3149	0	2.90×10^{-10}
Antixi +	31	1.32132	1	1.639×10^{-10}

Particle	No.	Mass(GeV)	Charge	Life time(sec)
Antiomega +	32	1.67245	1	8.22×10^{-11}
Deuteron	45	1.875613	1	stable
Tritium	46	2.80925	1	stable
Alpha	47	3.727417	2	stable
Geantino	48	0	0	stable
He3	49	2.80923	2	stable
Cerenkov	50	0	0	stable

`CALL` **GPIONS**

Stores the standard ions constants in the data structure `JPART`.

Particle	No.	Mass(GeV)	Charge	Life time(sec)
Li6	61	5.60305	3	1000
Li7	62	6.53536	3	1000
Be7	63	6.53622	4	1000
Be9	64	8.39479	4	1000
B10	65	9.32699	5	1000
B11	66	10.25510	5	1000
C12	67	11.17793	6	1000
N14	68	13.04378	7	1000
O16	69	14.89917	8	1000
F19	70	17.69690	9	1000
Ne20	71	18.62284	10	1000
Na23	72	21.41483	11	1000
Mg24	73	22.34193	12	1000
Al27	74	25.13314	13	1000
Si28	75	26.06034	14	1000
P31	76	28.85188	15	1000
S32	77	29.78180	16	1000
Cl35	78	32.57328	17	1000
Ar36	79	33.50356	18	1000
K39	80	36.29447	19	1000
Ca40	81	37.22492	20	1000
Sc45	82	41.87617	21	1000
Ti48	83	44.66324	22	1000
V51	84	47.45401	23	1000
Cr52	85	48.38228	24	1000
Mn55	86	51.17447	25	1000
Fe56	87	52.10307	26	1000
Co59	88	54.89593	27	1000
Ni58	89	53.96644	28	1000
Cu63	90	58.61856	29	1000
Zn64	91	59.54963	30	1000

Particle	No.	Mass(GeV)	Charge	Life time(sec)
Ge74	92	68.85715	32	1000
Se80	93	74.44178	34	1000
Kr84	94	78.16309	36	1000
Sr88	95	81.88358	38	1000
Zr90	96	83.74571	40	1000
Mo98	97	91.19832	42	1000
Pd106	98	98.64997	46	1000
Cd114	99	106.10997	48	1000
Sn120	100	111.68821	50	1000
Xe132	101	122.86796	54	1000
Ba138	102	128.45793	56	1000
Ce140	103	130.32111	58	1000
Sm152	104	141.51236	62	1000
Dy164	105	152.69909	66	1000
Yb174	106	162.02245	70	1000
W184	107	171.34924	74	1000
Pt194	108	180.67513	78	1000
Au197	109	183.47324	79	1000
Hg202	110	188.13451	80	1000
Pb208	111	193.72907	82	1000
U238	112	221.74295	92	1000

Note

It is possible for the user to define more particles or to redefine some characteristics of the particles currently defined in `GEANT`, but this must be done with extreme care. In particular, the mass and charge of most particles are stored independently in `FLUKA`, and any change made via `GSPART` will not affect these values. Removing particles from the list can lead to unpredictable results and it is strongly discouraged.

The user who needs more particles, or wants to partly override the standard values, can do that via the routines `GSPART` and `GSDK`.

All data taken from M. Aguilar Benitez [4] and updated with the values of the PDG [5].

`CALL` **GSPART** `(IPART,CHNPAR,ITRTYP,AMASS,CHARGE,TLIFE,UB,NWB)`

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

`IPART` (`INTEGER`) particle number;

`CHNPAR` (`CHARACTER*20`) particle name;

`ITRTYP` (`INTEGER`) type of tracking routine requested:

- `1` particle tracked by `GTGAMA`;
- `2` particle tracked by `GTELEC`;
- `3` particle tracked by `GTNEUT`;
- `4` particle tracked by `GTHADR`;
- `5` particle tracked by `GTMUON`;
- `6` geantino tracked by `GTNINO`;
- `7` heavy ion tracked by `GTCKOV`;
- `8` light photon tracked by `GTHION`;

`AMASS` (`REAL`) particle mass in GeV;

`CHARGE` (`REAL`) particle charge;

TLIFE	(REAL) particle life time (in seconds);
UB	(REAL) array of NWB user additional parameters;
NWB	(INTEGER).

```
CALL GFPART (IPART,CHNPAR*,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 for particle IPART (for all particles if IPART=0).

Origin : R.Brun, G.N.Patrick **Submitted:** 23.01.84
Revision : **Revised:** 19.10.94
Documentation :

Branching Ratios and Particle Decay Modes

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

`IPART` (`INTEGER`) `GEANT` particle number;
`BRATIO` (`REAL`) array of branching ratios in %, maximum 6;
`MODE` (`INTEGER`) array of partial decay modes.

`GSDK` stores the branching ratios and partial decay modes for two and three body particle decays into the data structure `JPART`. The decay modes should be coded into the array `MODE` such that:

$$\texttt{MODE(I)} = 10^4\ \texttt{N3} + 10^2\ \texttt{N2} + \texttt{N1} \qquad \text{for } \texttt{I=1,}\cdots\texttt{,6}$$

where:

`N1` particle number for decay product 1;
`N2` particle number for decay product 2;
`N3` particle number for decay product 3 (if any).

It is important to note the following:

- prior to calling `GSDK`, all parent and secondary particles should have been defined by `GSPART`;
- if less than six decay modes are defined the remaining elements of `BRATIO` and `MODE` must be set 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 the bank containing the information on the particle `IPART`;
`JDK1` = `LQ(JPA-1)` pointer to branching ratios bank;
`JDK2` = `LQ(JPA-2)` pointer to decay modes bank;
`BR(I)` = `Q(JDK1+I)` I^{th} branching ratio;
`MODE(I)` = `IQ(JDK2+I)` I^{th} decay mode, where `I=1,`$\cdots$`,6`.

When a particle decays during tracking, the routine `GDECAY` is called. If branching ratios and modes have been stored by `GSDK`, then `GDECAY` generates the decay products in the 2- or 3-body phase space. If no branching ratios have been defined by `GSDK`, the user routine `GUDCAY` is called, where the user can code special decay modes and branching ratios. All data taken from Aguilar Benitez [4].

Parent particle	Decay	Branching ratio (%)
$\mu^\pm$	$e^\pm \nu\nu$	100.00
π^0	$\gamma\gamma$	98.802
	$\gamma e^+ e^-$	1.198
$\pi^\pm$	$\mu^\pm \nu$	100.00
K_l^0	$\pi^- e^+ \nu$	19.35
	$\pi^+ e^- \nu$	19.35
	$\pi^- \mu^+ \nu$	13.50
	$\pi^+ \mu^- \nu$	15.50
	$\pi^0\pi^0\pi^0$	21.50
	$\pi^+\pi^-\pi^0$	12.38
$K^\pm$	$\mu^\pm \nu$	63.51
	$\pi^\pm\pi^0$	21.17
	$\pi^\pm\pi^+\pi^-$	5.59
	$e^\pm \nu \pi^0$	4.82
	$\mu^\pm \nu \pi^0$	3.18
	$\pi^\pm\pi^0\pi^0$	1.73
K_s^0	$\pi^+\pi^-$	68.61
	$\pi^0\pi^0$	31.39
η	$\gamma\gamma$	38.80
	$\pi^0\pi^0\pi^0$	31.90
	$\pi^+\pi^-\pi^0$	23.60
	$\pi^+\pi^-\gamma$	4.88
	$e^+e^-\gamma$	0.50
	$\pi^0\gamma\gamma$	0.071
Λ	$p\pi^-$	63.90
	$n\pi^0$	35.80
Σ^+	$p\pi^0$	51.57
	$n\pi^+$	48.30
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^-	67.80
	$\Xi^0\pi^-$	23.60
	$\Xi^-\pi^0$	8.60
$\bar{\Lambda}$	$\bar{p}\pi^+$	63.90
	$\bar{n}\pi^0$	35.80
$\bar{\Sigma}^-$	$\bar{p}\pi^0$	51.57
	$\bar{n}\pi^-$	48.30
$\bar{\Sigma}^0$	$\Lambda\gamma$	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

Parent particle	Decay	Branching ratio (%)
$\bar{\Omega}^+$	$\bar{\Lambda}K^+$	67.80
	$\bar{\Xi}^0\pi^+$	23.60
	$\bar{\Xi}^+\pi^0$	8.60

CONS Bibliography

[1] Particle Data Group. Review of particle properties. *Physical Review D*, 45(11), June 1992.

[2] R.Ford and W.Nelson. SLAC 210, UC 32. SLAC, June 1978.

[3] J.Allison. Calculation of hadronic interaction lengths for mixtures. Technical Report OPAL/0037N/JA/md, CERN, Hadron Milestone 84/003 1984.

[4] Aguillar Benitez et al. Review of particle properties. *Rev. Mod. Phys.*, 56(198), 1984.

[5] Particle Data Group. Review of particle properties. *Physical Review D*, 50(1173), 1994.

Origin :
Revision :
Documentation : P.Zanarini, S.Giani

Submitted: 07.03.92
Revised: 11.12.92

Introduction to the Drawing package

1 The drawing package

The drawing package has been designed mainly to visualise:

- detector components;
- the logical tree of detector components;
- the geometrical dimensions;
- particle trajectories
- the hits recorded in the sensitive elements of the detector.

2 Graphics in GEANT

The set of routines in the drawing package allows the visualisation of the volumes of the detector. These routines can be called from the user program or interactively via the command interface. The output can be directed onto the screen or to a file. The graphics of GEANT is based on the HIGZ [1] package. HIGZ is a high-level interface between the user program (in this case GEANT), ZEBRA and a basic graphics library. At the moment of writing, HIGZ can use various flavours of the GKS [2, 3, 4] graphics system or the X11 [5] system. A version based on the PHIGS [6] system is now being developed at CERN.

3 Graphics functionality of GEANT

The graphics capabilities of GEANT have been greatly enhanced in the last few revisions of the program. It is now possible to display volumes with hidden lines removed from the drawing. This improves the understanding of complicated objects. It is also possible to *cut* the displayed volumes with various shapes (boxes, cones, tubes and spheres) in order to visualise the internals of a detector element. The surfaces of the volumes drawn can be filled with colour, either solid or shaded according to a scan-line based lighting algorithm. A more sophisticated surface rendering algorithm is now being developed but it will be available only in the next version of the program.

Almost all drawing commands can be executed either by calling a routine from the user application or by issuing a command when running GEANT with the standard interactive interface. Several interactive tools (ZOOM, LENS) have been developed to analyse in detail the various parts of the detectors. The interactive MOVE facility allows the rotation, translation and zooming in real time for simple components of the detectors. Some of these commands are only available in the interactive interface. These are described briefly in [DRAW500] and more extensively in the [XINT] section.

A user interface based on X-Windows and Motif is also available mainly on Unix workstations. Details of this interface are described in [DRAW510].

Various graphical options can be set via the GSATT and GDOPT routines.

Every detector component can be visualised either in orthonormal projection or in perspective (routines GDRAW and GDRVOL). Sections of the volumes and of their descendents can be visualised as well (routines GDRAWX and GDRAWC).

The hierarchical relations of the various volumes which compose the detector can be visualised via the GDTREE routine. In the interactive interface the user can select one of the volumes which are nodes of the tree with the mouse, and display the characteristics of the volume and walk up and down the logical tree.

Tracks and hits can be drawn either alone or superimposed on the detector (routines GDCHIT, GDHITS, GDAHIT, GDXYZ). It is also possible to draw the tracks while tracking is performed via the routine GDCXYZ.

The drawing may be done with hidden-line removal and with shading effects according to the value of the options HIDE and SHAD (see next section).

Drawing complicated detectors can be a lengthy process even on a fast machine. In particular, when graphics options like hidden line removal or surface shading are requested and many volumes are involved, the complete drawing can take several minutes on the most powerful workstations. To alleviate this problem drawings can be stored in memory in so called *view banks*. The routines involved are GDOPEN, GDCLOS, GDSHOW and GDELET, and the corresponding interactive commands (DOPEN, DCLOSE, DSHOW, DELETE). These banks can then be redisplayed quickly. They can also be written on disk and read back.

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.

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.

The geometrical dimensions of an existing volume can be changed via the routine GEDITV.

Various service routines allow the drawing of the profile of a human figure, the axis, the scale of the dimension of the drawing and so on. These are described in [DRAW400].

4 The shading algorithm

In GEANT 3.16 a new shading algorithm has been introduced which is 10 to 100 times faster than the algorithm which was implemented in GEANT 3.15. In the new algorithm the search for the visible portions of all the faces of every volume is performed in the following steps:

1. the vertices of each face are transformed from the MARS to the screen reference system. In this way the dimensions of the face are of the order of magnitude of centimeters.
2. in the screen reference system the face is *filled* with horizontal lines *scan-lines* according to the desired resolution.
3. These lines are then transformed back in the MARS, where the hidden line removal technique is used to find their visible portions.
4. The colour luminosity is chosen as a function of the angle between the normal to the surface and the line of view which is considered to be also the line of illumination.
5. The visible portions of the lines are finally transformed into the screen reference system and drawn with the calculated luminosity so that they fill only the visible part of each face.

The resulting picture can be saved into GEANT view banks or HIGZ picture files.

5 Summary

The best way to initialise the drawing package is to call the HPLOT [7] initialisation routine HPLINT. This will work independently of the basic graphic package used. The reader should consult the HPLOT documentation for more information on this routine. The drawing package is initialised by:

HPLINT to initialise the basic graphics package and the HPLOT package;
GDINIT to initialise the GEANT drawing package.

The 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 modelled 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 the 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
`GDLENS` to scan interactively the drawing contained in the view bank
`GDELET` to delete a given view bank from memory

Routines that allow the visualisation of the detector with hidden line removal are:

`GDCGOB` to create CG objects
`GDCGSL` to create clipping objects
`GDCGCL` to perform boolean operations
`GDCGHI` to insert CG objects in HIDE and WIRE structures of faces
`GDSHIF` to shift in the space a CG object
`GDBOMB` to make the detector explode

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.

Origin : S.Giani **Submitted:** 10.03.94
Revision : **Revised:** 10.03.1994
Documentation : S.Giani, S.Ravndal

The Ray-tracing package

Running interactively under X11, a new visualization tool is provided in GEANT 3.21: the ray-tracing package. Based on the new GEANT tracking (see GEOM and TRACK), a set of routines doing light processing is provided to visualize the detectors (useful also to visualize the results of boolean operations). Basically, visible light particles are tracked throughout the detector until when they do not hit the surface of a volume declared not transparent; then, the intersection point is transformed to the screen coordinates and the corresponding pixel is drawn with a computed hue and luminosity.

In case the command (DOPT RAYT ON) is executed, the drawings are performed by the GEANT ray-tracing; automatically, the color is assigned according to the tracking medium of each volume and the volumes with a density lower/equal than the air are considered transparent; if the option (USER) is set (ON) (again via the command (DOPT)), the user can set color and visibility for the desired volumes via the command (SATT), as usual, relatively to the attributes (COLO) and (SEEN). The resolution can be set via the command (SATT * FILL VALUE), where (VALUE) is the ratio between the number of pixels drawn and 20 (user coordinates). Parallel view and perspective view are possible (DOPT PROJ PARA/PERS).

In the first case, we assume that the first mother volume of the tree is a box with dimensions 10000 X 10000 X 10000 cm and the view point (infinetely far) is 5000 cm far from the origin along the Z axis of the user coordinates; in the second case, the distance between the observer and the origin of the world reference system is set in cm by the command (PERSP NAME VALUE). Grand-angle or telescopic effects can be achieved changing the scale factors in the command (DRAW). (Please, note that in case of perspective views, fixed the distance of the viewer, the zooming factors for the raytracing option must be 8.25 times bigger than the ones used for the HIDE options in order to get the same picture). When the final picture does not occupy the full window, mapping the space before tracing can speed up the drawing, but can also produce less precise results; values from 1 to 4 are allowed in the command (DOPT MAPP VALUE), the mapping being more precise for increasing (VALUE). For (VALUE = 0) no mapping is performed (therefore max precision and lowest speed).

The command (VALCUT) allows the cutting of the detector by three planes ortogonal to the x,y,z axis. The attribute (LSTY) can be set by the command SATT for any desired volume and can assume values from 0 to 7. It determines the different light processing to be performed for different materials: 0 = dark-matt, 1 = bright-matt, 2 = plastic, 3 = ceramic, 4 = rough-metals, 5 = shiny-metals, 6 = glass, 7 = mirror.

The detector is assumed to be in the dark, the ambient light luminosity is 0.2 for each basic hue (the saturation is 0.9) and the observer is assumed to have a light source (therefore he will produce parallel light in the case of parallel view and point-like-source light in the case of perspective view). Finally, a second light source can be positioned in the space (with a desired intensity) via the command (SPOT).

Origin : R.Brun, A.McPherson, P.Zanarini
Revision : S.Giani
Documentation : P.Zanarini, S.Giani

Submitted: 15.08.83
Revised: 11.12.92

Drawing a Volume – Case 1

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

Draws an orthographic parallel projection or a perspective projection of volume `CHNAME` according to the options set via the routine `GDOPT` and all its *visible* descendants (see routine `GSATT` to alter the visibility of a volume) at 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 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 the common `/GCDRAW/`. Attributes like colour, surface fill, line width, line style, visibility, etc. can be set by calling the `GSATT` routine for `CHNAME` and its descendants `[GEOM 500]`. An example of the result of the call:

```
CALL GDRAW('ARM ',40.,40.,0.,nx,ny,0.015,0.015)
```

is given in fig 8 for various values of the attributes and graphic options.

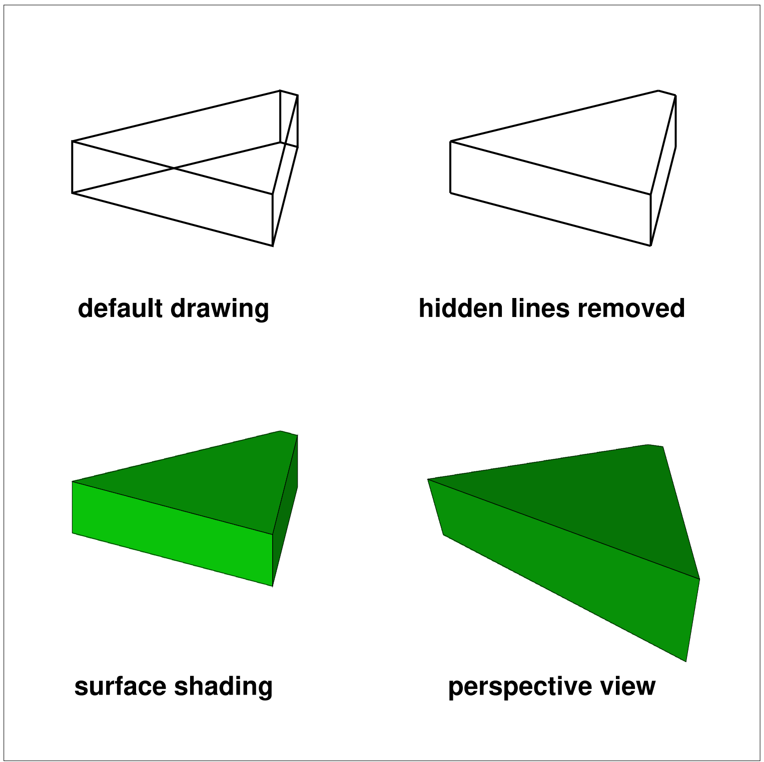


Figure 8: Examples of use of `GDRAW`

Drawing with hidden line removal and surface fill shading is also possible, but the drawing time may increase substantially for complicated objects.

When using the interactive interface it is possible to *cut* the object to be drawn with *cutting* volumes (command `CVOL`). This is particularly useful to visualise the contents of an object drawn with hidden line removal. Another feature which is available only interactively is the possibility to draw each decendant of a volume shifted along the axis which goes from the centre of the local coordinate system to the centre of gravity of the descendant itself. This again is useful to improve the visibility of the details of a complicated setup (command BOMB). Please see the interactive interface description for more information on this part.

CHNAME (CHARACTER*4) volume name.
THETA (REAL) theta angle between the line of sight and the Z axis of MAster Reference System (MARS).
PHI (REAL) phi angle between the projection of the line of sight on plane X Y and the X axis of MARS.
PSI (REAL) psi angle by which the projected image will be rotated on the screen plane.
U0 (REAL) u coordinate on the screen of the volume origin.
V0 (REAL) v coordinate on the screen of the volume origin.
SU (REAL) scale factor for u coordinates.
SV (REAL) scale factor for v coordinates.

Origin : P.Zanarini **Submitted:** 01.01.86
Revision : **Revised:** 11.12.92
Documentation : P.Zanarini, F.Carminati

Drawing a Volume Projection view – Case 2

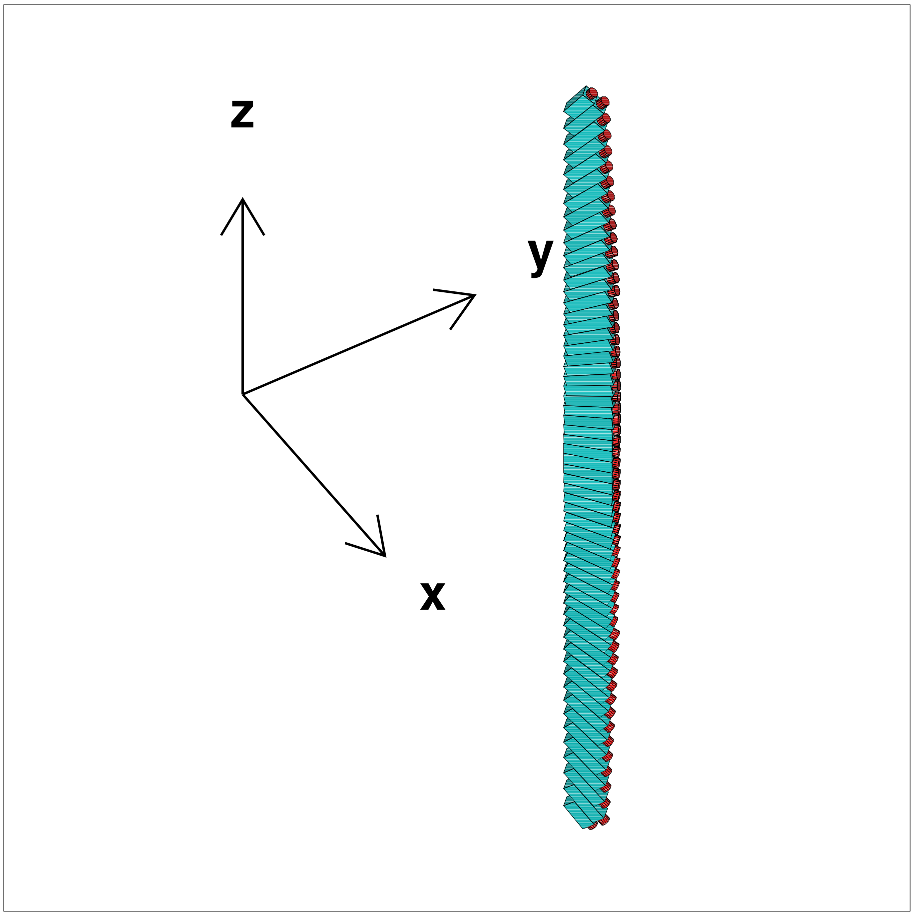
```
CALL GDRVOL (NNAMS,CHNAMS,LNUMBS,NRS,THETA,PHI,PSI,U0,V0,SU,SV)
```

Draws an orthographic parallel projection or a perspective projection (depending on the option chosen via GDOPT) of the volume CHNAMS(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 by the GSATT routine for CHNAMS(N) and or its descendants [GEOM 500]. This routine differs from GDRAW in the following aspects:

- The object to be drawn is identified by a full path. This gives the possibility of drawing a particular copy or division of a volume, or even a volume that has more than one mother in the geometry tree. CHNAMS(1),...,CHNAMS(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 in its position 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 is done normally (to reset to the normal case a call with NRS=0 or NNAMS=0 is required).

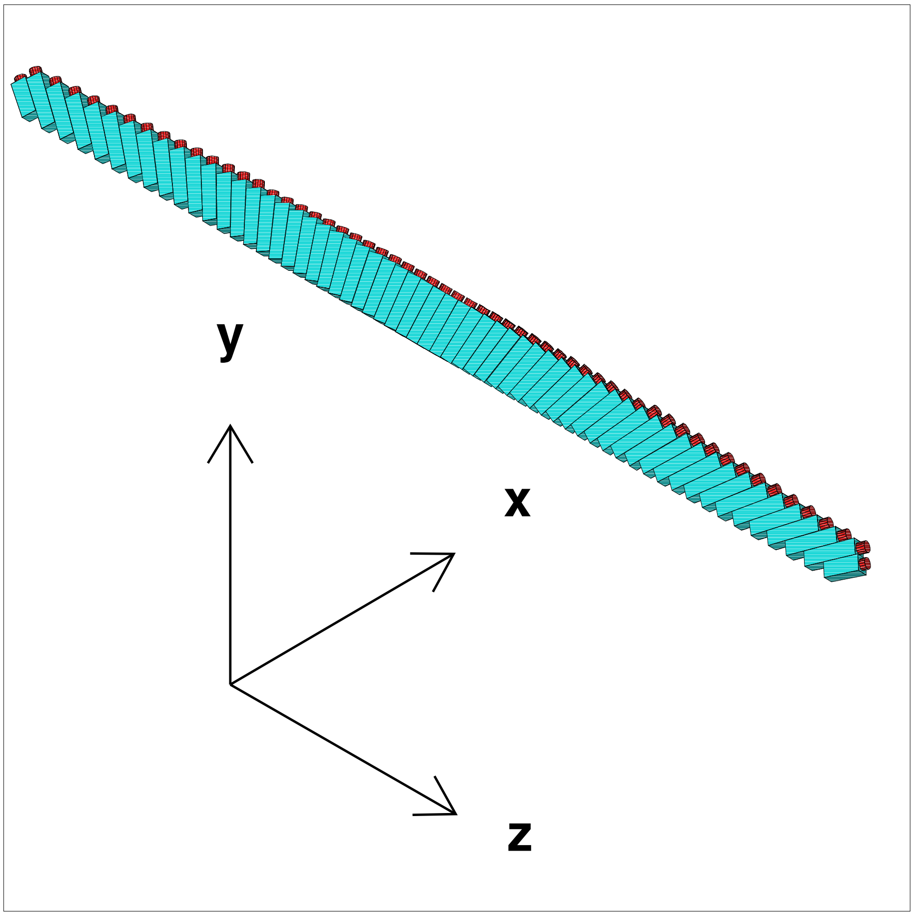
NNAMS (INTEGER) number of elements levels in the arrays CHNAMS, LNUMBS. The bottom volume of this path is also the one that is actually drawn;
CHNAMS (CHARACTER*4) array of volume name (dimensioned at least to NNAMS);
LNUMBS array of volume numbers (dimensioned at least to NNAMS);
NRS reference system 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 θ angle between the line of sight and the Z axis of MARS;
PHI ϕ angle between the projection of the line of sight on plane X Y and the X axis of MARS
PSI ψ 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

Two examples of use of GDRVOL are shown in fig 9 and 10.




```
        CHARACTER*4 CHNAMS(5)
        INTEGER LNUMBS(5)
        DATA CHNAMS/'OPAL','BRL-','EB  ','EBB ','EBP '/
        DATA LNUMBS/    1 ,    1 ,    1 ,    1 ,   20 /
        .
        .
        .
        NRS=0
        CALL GDRVOL(5,CHNAMS,LNUMBS,NRS,80.,135.,0.,13.,10.,0.03,0.03)
        CALL GDAXIS(0.,0.,0.,200.)
C       CALL GDXYZ(0)
```

Figure 9: Example of use of `GDRVOL` in the MAster Reference System




```
      CHARACTER*4 CHNAMS(5)
      INTEGER LNUMBS(5)
      DATA CHNAMS/'OPAL','BRL-','EB  ','EBB ','EBP '/
      DATA LNUMBS/    1 ,    1 ,    1 ,    1 ,   20 /
      .
      .
      .
      NRS=1
      CALL GDRVOL(5,CHNAMS,LNUMBS,NRS,55.,135.,0.,5.,5.,0.035,0.035)
      CALL GDAXIS(0.,0.,0.,200.)
C     CALL GDXYZ(0)
```

Figure 10: Example of use of `GDRVOL` in the Daughter Reference System

Origin : P.Zanarini
Revision :
Documentation : P.Zanarini

Submitted: 15.05.84
Revised: 10.12.93

Draw a volume cut view

```
CALL GDRAWX (CHNAME,CUTTHE,CUTPHI,CUTVAL,THETA,PHI,UO,VO,SU,SV)
```

CHNAME	(CHARECTER*4) name of volume;
CUTTHE	(REAL) θ angle of the line normal to the cut plane;
CUTPHI	(REAL) ϕ angle of the line normal to the cut plane;
TETHA	(REAL) viewing angle θ;
PHI	(REAL) viewing angle ϕ;
UO	(REAL) u coordinate on the screen of the volume origin;
VO	(REAL) v coordinate on the screen of the volume origin;
SU	(REAL) scale factor for u coordinates;
SV	(REAL) scale factor for v coordinates.

Draws a *cut view* of the volume CHNAME with all its *visible* descendants, i.e. draws their intersection with the cut plane normal to the axis given by the angles CUTTHE, CUTPHI at the distance CUTVAL from the origin. The view point is defined by the angles THETA, PHI. UO, VO, SU, SV have the same meaning as in GDRAW. These *view parameters*, as well as zoom parameters set by GDZOOM are copied in /GCDRAW/.

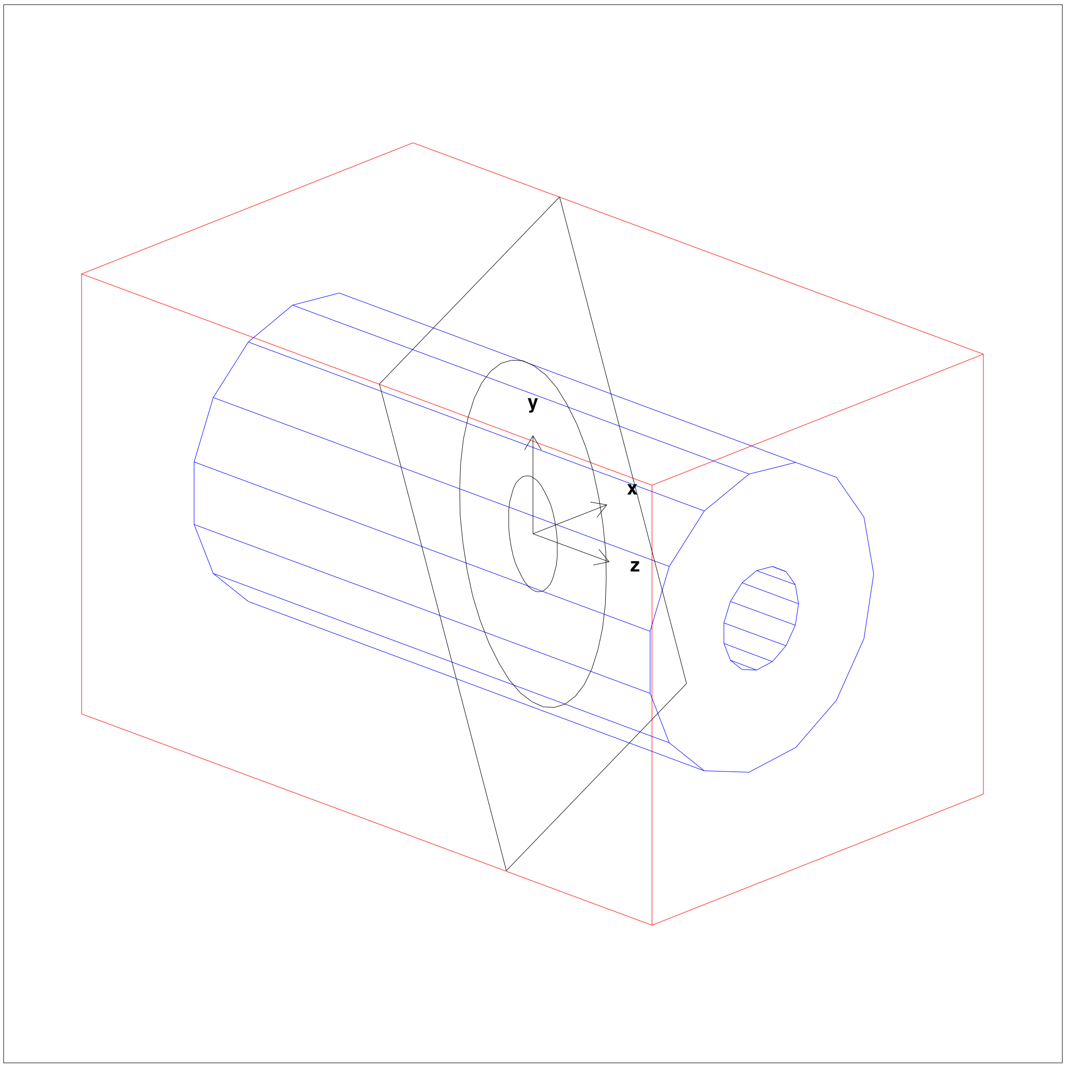
Attributes like colour, surface fill, line width, line style, visibility, etc. can be set by the GSATT routine for CHNAME and its descendants [GEOM500].

```
CALL GDRAWC (CHNAME,IAX,CUTVAL,UO,VO,SU,SV)
```

CHNAME	(CHARECTER*4) name of volume;
IAX	(INTEGER) axis of the cut (1=X, 2=Y, 3=Z);
UO	(REAL) u coordinate on the screen of the volume origin;
VO	(REAL) v coordinate on the screen of the volume origin;
SU	(REAL) scale factor for u coordinates;
SV	(REAL) scale factor for v coordinates.

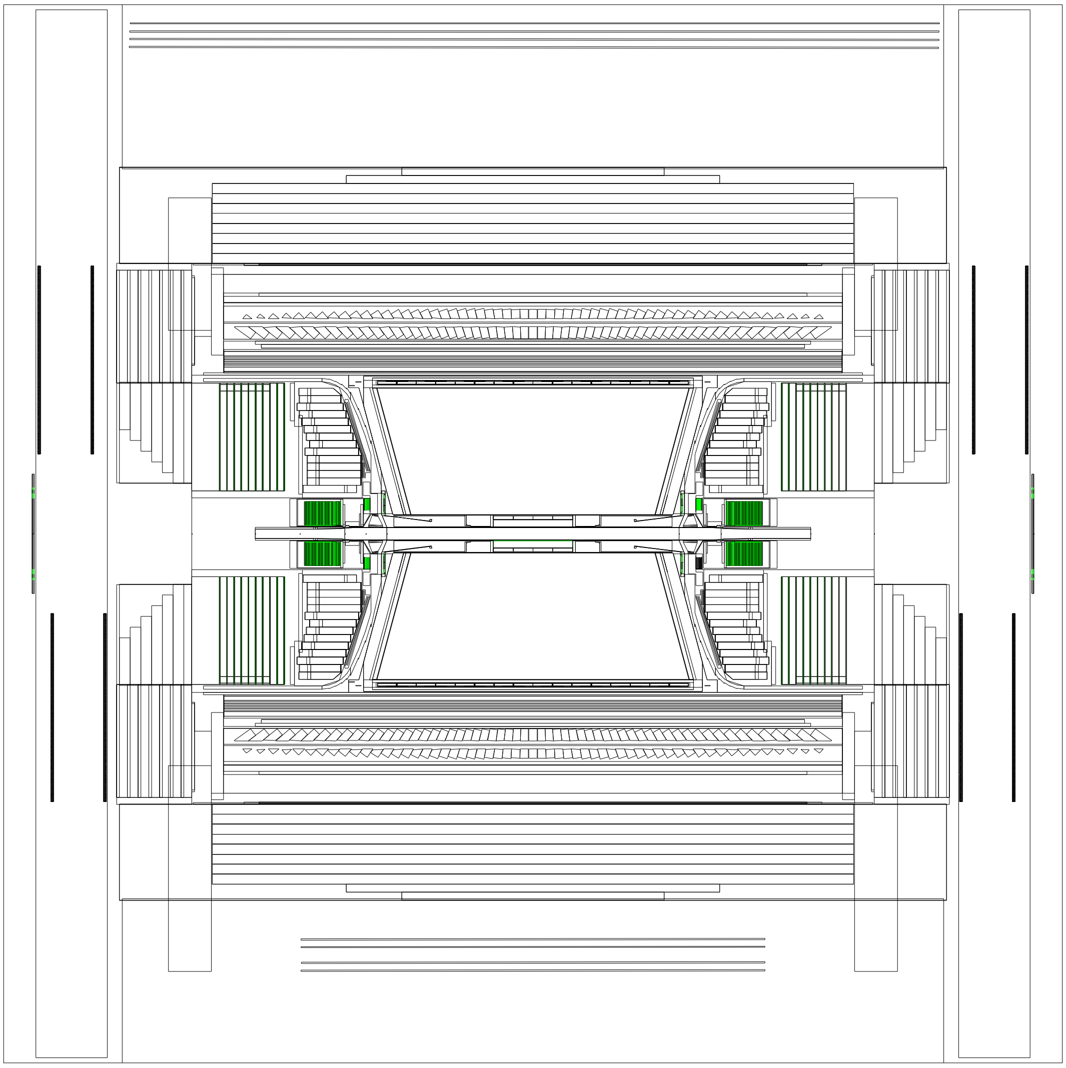
This routine is a special case of the previous GDRAWX. Here the cut plane is normal to one of the main axes (IAX) and placed at a distance CUTVAL from the origin. The view-point is along the same axis.

Fig. 11 gives an example of utilisation of GDAXIS, GDRAW and GDRAWX. Fig. 12 gives an example of use of GDRAWC.




```
CALL GDOPT('HIDE','ON')
CALL GSATT('NBOX','COLO',2.)
CALL GSATT('NTUB','COLO',4.)
CALL GDRAW('NBOX',50.,150.,0.,10.,10.,0.01,0.01)
CALL GDRAW('NTUB',50.,150.,0.,10.,10.,0.01,0.01)
CALL GDOPT('HIDE','OFF')
CALL GSATT('*','COLO',1.)
CALL GDRAWX('NBOX',30.,40.,0.,50.,150.,10.,10.,0.01,0.01)
CALL GDAXIS(0.,0.,0.,200.)
```

Figure 11: Example of utilisation of GDAXIS, GDRAW and GDRAWX



```
CALL GDRAWC('OPAL',2,0.,10.,10.,0.01,0.01)
```

Figure 12: Example of utilisation of GDRAWC

Origin : R.Brun, P.Zanarini
Revision :
Documentation : P.Zanarini

Submitted: 15.05.84
Revised: 11.12.92

Draw Particle Trajectories

1 Visualisation of particle trajectories

The tracks generated by the tracking package, and optionally stored in the data structure JXYZ, can be displayed by the routine GDXYZ (corresponding to the interactive command DXYZ).

The names of the particles and or the track numbers can be added by the routine GDPART (corresponding to the interactive command DPART).

A special routine has been provided to visualise the tracks during the transport process (GDCXYZ), that could be called for instance by GUSTEP. That routine shows the tracks while they are transported by the tracking package of GEANT, providing a useful interactive debugging tool.

```
CALL GDXYZ (ITRA)
```

Draws track number ITRA for which space points have been stored in bank JXYZ via calls to the routine GSXYZ. The view parameters are taken from /GCDRAW/.

ITRA (INTEGER) 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	- - - - - - - - - -
(yellow)	dotted line for Čerenkov photons	....

```
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 the bank JXYZ via the routine GSXYZ. At present only primary tracks are displayed by GDPART and their name or number is written at the end the track trajectory. The view parameters are taken from /GCDRAW/.

ITRA (INTEGER) Track number (if 0 all tracks are taken)
ISEL (INTEGER)
ISEL=x1 draws the track number,
ISEL=1x draws the particle name,
ISEL=11 draws both.
SIZE (REAL) Character size in cm.

```
CALL GDCXYZ
```

If GDCXYZ is called at tracking time (for instance by GUSTEP), it draws the tracks while tracking is performed, at the same time as they are generated by the tracking package of GEANT. This is a very useful interactive debugging tool. The line style is the same as for GDXYZ. The view parameters are taken from /GCDRAW/.

Here we give an example of the use of GDPART:

```
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)
```

Origin : R.Brun, W.Gebel, P.Zanarini
Revision :
Documentation : S.Giani, F.Carminati

Submitted: 07.03.84
Revised: 11.12.92

Drawing Track Hits in Sensitive Detectors

The hits generated by the tracking package and stored in the data structure `JHITS` can be displayed by the following routines:

1. draw one hit (`GDAHIT`); called by the user;
2. draw all the hits of trajectory type sets/ detectors (`GDHITS`);
3. draw all the hits of calorimeter type sets/ detectors (`GDCHIT`).

Different symbols for each 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` and `GDHITS`, while it is computed as a function of the hits value in `GDCHIT`. The option `H` of the interactive `MOVE` command gives the possibility to rotate, zoom and translate in real time the hits stored for one event. This allow a 3D display of the event where it is possible to pick one hit with the mouse and have the hit values printed. The interactive command `LENS` allows a detailed visual scanning of the hits displayed (see the interactive section).

`CALL` **GDHITS** `(CHUSET,CHUDET,ITRA,ISYMB,SSYMB)`

Draws the hit points as stored by `GSAHIT`, which were generated by track `ITRA` in detector `CHUDET` of set `CHUSET`, with the currently selected view parameters in `/GCDRAW/`. The character plotted at each point may be chosen via `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)

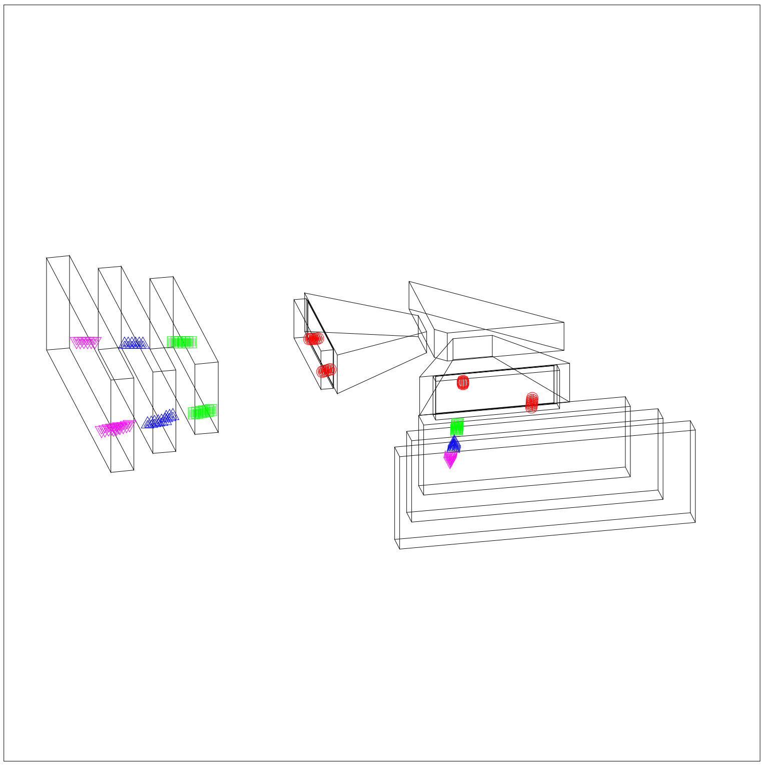
More information on the meaning of the `ISIMB` value can be found in the `HPLOT` manual [7]. Any value for `ISYMB` which is not in the above table is equivalent to option 0. Except for `ISYMB = -1`, the size of the character can be chosen by `SSYMB`. On the 2D projection on the screen one can distinguish 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 suit 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 `[HITS200]`.

`CHUSET` (`CHARACTER*4`) user set identifier (if '*' all sets are taken)
`CHUDET` (`CHARACTER*4`) user detector identifier (if '*' all detectors are taken)
`ITRA` (`INTEGER`) number of selected track (if 0 all tracks are taken)
`ISYMB` (`INTEGER`) character selection number (see table above)
`SSYMB` (`REAL`) size of characters in cm (if 0, a default of 0.1 is taken)

`CALL` **GDCHIT** `(CHUSET,CHUDET,ITRA,ISYMB,SIZMAX,IHIT,HITMIN,HITMAX)`

Draws the hit points as stored by `GSCHIT`, which were generated by track `ITRA` in detector `CHUDET` of set `CHUSET`, 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)`:



```
CALL GDRAW('CAVE',40.,40.,0.,10.,10.,.04,.04)
CALL IGSET('TXCI',2.)
CALL GDHITS('DRFT','FSP ',0,850,.3)
CALL IGSET('TXCI',3.)
CALL GDHITS('DRFT','RSP1',0,851,.3)
CALL IGSET('TXCI',4.)
CALL GDHITS('DRFT','RSP2',0,852,.3)
CALL IGSET('TXCI',5.)
CALL GDHITS('DRFT','RSP3',0,853,.3)
```

Figure 13: example of use of GDHITS

```
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 [HITS200].

SIZMAX (REAL) maximum character size in cm

IHIT (INTEGER) HITS(IHIT) contains the calorimeter hit value. IHIT>3, and the first three elements are supposed to contain the space position of the hit

HITMIN (REAL) minimum boundary of HITS(IHIT)

HITMAX (REAL) maximum boundary of HITS(IHIT)

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

Draws one hit point at coordinates X, Y, Z. This routine is useful when the user wants only to display but not to store the hits. The first three arguments are the position of the hit in the MARS, and the last two arguments have the same meaning as the similar arguments in GDHITS.

X (REAL) x coordinate in MARS of the hit point

Y (REAL) y coordinate in MARS of the hit point

`X` (`REAL`) z coordinate in `MARS` of the hit point

An example of use of the hits drawing routines is given in fig 13.

Origin : P.Zanarini
Revision : S.Giani
Documentation : P.Zanarini, S.Giani

Submitted: 15.01.84
Revised: 10.12.93

Drawing the geometrical tree

The geometrical tree is a hierarchical representation of the structure of the detector, based on the mother-daughter relationship among the various volumes composing it. The GDTREE routine (corresponding to the DTREE interactive command) allows the drawing of the logical tree, displaying name, multiplicity and other information about the volumes.

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

CHNAME (CHARACTER*4) volume name;

LEVMAX (INTEGER) depth levels (if 0 all levels are taken) default is 3;

ISEL (INTEGER) used to select options in the picture of the tree:

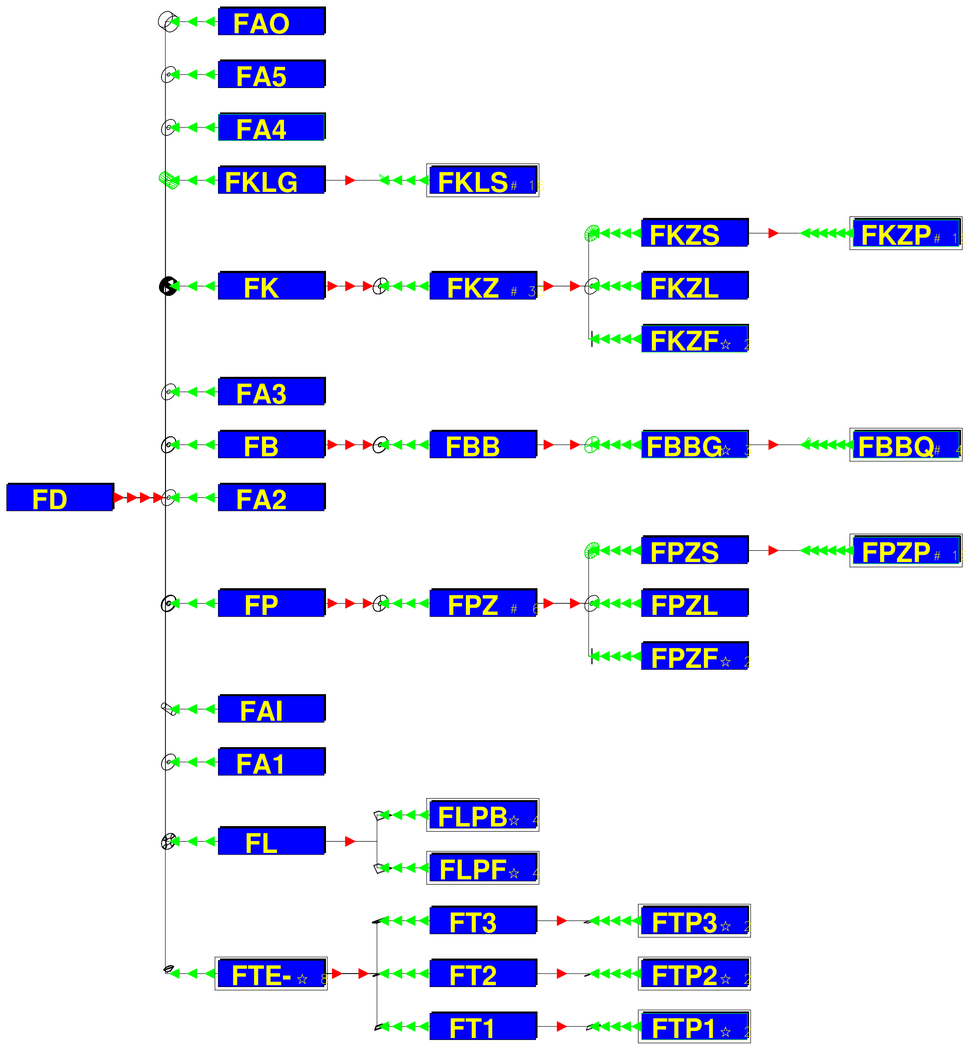
- 0 only the node name is drawn;
- xxxx1 multiplicity is added:
  - each node that has been positioned several times (by GSPOS or GSPOSP) has a multiplicity number preceded by the symbol '*';
  - each node that has been obtained by division mechanism (by GSDVN, GSDVT, etc.) has a multiplicity number preceded by the symbol #;
- xxx1x ONLY information is added. volumes that are MANY will be drawn as three square nodes, one overlapping the others but slightly shifted;
  Note: if CHNAME itself is MANY it will not be recognised as such, because this information is stored in its mother which is unknown;
- xx1xx DET information is added; if a volumes is a detector it is drawn as a square node with another one slightly larger around it;
- x1xxx SEEN information is added; if a volume is not seen (either because of its SEEN attribute or because of its ancestors SEEN attributes affecting the whole tree path) its square node is drawn by using dashed lines;
- 1xxxx a little picture of the volume is added above each node.

GDTREE draws the tree of the volume data structure starting from the volume CHNAME, for LEVMAX depth levels; a continuation link line is drawn for those elements at the last level on the picture but having descendants. An example of use of GDTREE can be found in Fig. 14. For every node, as many red arrows as the number of existing levels below it are plotted on the link; vice versa, as many green arrows as the number of existing levels before it are also plotted. When running GEANT interactively, if the ISEL argument is not specified, control is given to the mouse after issuing the DTREE command. This allows the display of various volume parameters and the navigation of the geometrical tree by clicking the mouse buttons. The size of the picture and the dimensions of each element and arrows are computed automatically.

Interactive use

In an interactive session, control can be given to the mouse:

- clicking on the left button when the cursor is inside a volume *pave* will generate a DSPEC command for that volume;
- clicking the same button when the cursor is on a red arrow will perform a DTREE for the volume preceding the arrow, displaying a number of levels which depends on the which arrow has been selected;




```
CALL GDTREE('FD  ',5,11111)
```

Figure 14: Example of geometrical tree structure

- selecting in the same way the ith green arrow preceding a given volume will perform a DTREE for its mother-volume staying *i* levels before;
- clicking the right button of the mouse will return the control to the command mode.

When running with the X-windows software, the drawing of the specification (DSPEC) appears in a different window to speed up the scanning of the tree. Iterating this procedure it is possible to analyse very easily and quickly any kind of geometry. This drawing tree capability can be very useful when tuning the detector geometry. See the interactive section [XINT] for examples.

Origin : P.Zanarini **Submitted:** 01.11.83
Revision : **Revised:** 13.12.93
Documentation : P.Zanarini, S.Giani, F.Carminati

Drawing volume specifications

The geometrical parameters of the volumes can be displayed via the GDSPEC routine (corresponding to the DSPEC interactive command). This facility provides a detailed picture of a particular piece of the detector. The set of geometrical specifications of all the descendants of a given node on the tree, can be obtained each on a separate picture with the routine GDFSPC (DFSPC interactive command).

```
CALL GDSPEC (CHNAME)
```

CHNAME (CHARACTER*4) volume name;

Draws a picture showing all specifications for a given volume. An example of use of GDSPEC can be found in Fig. 15.

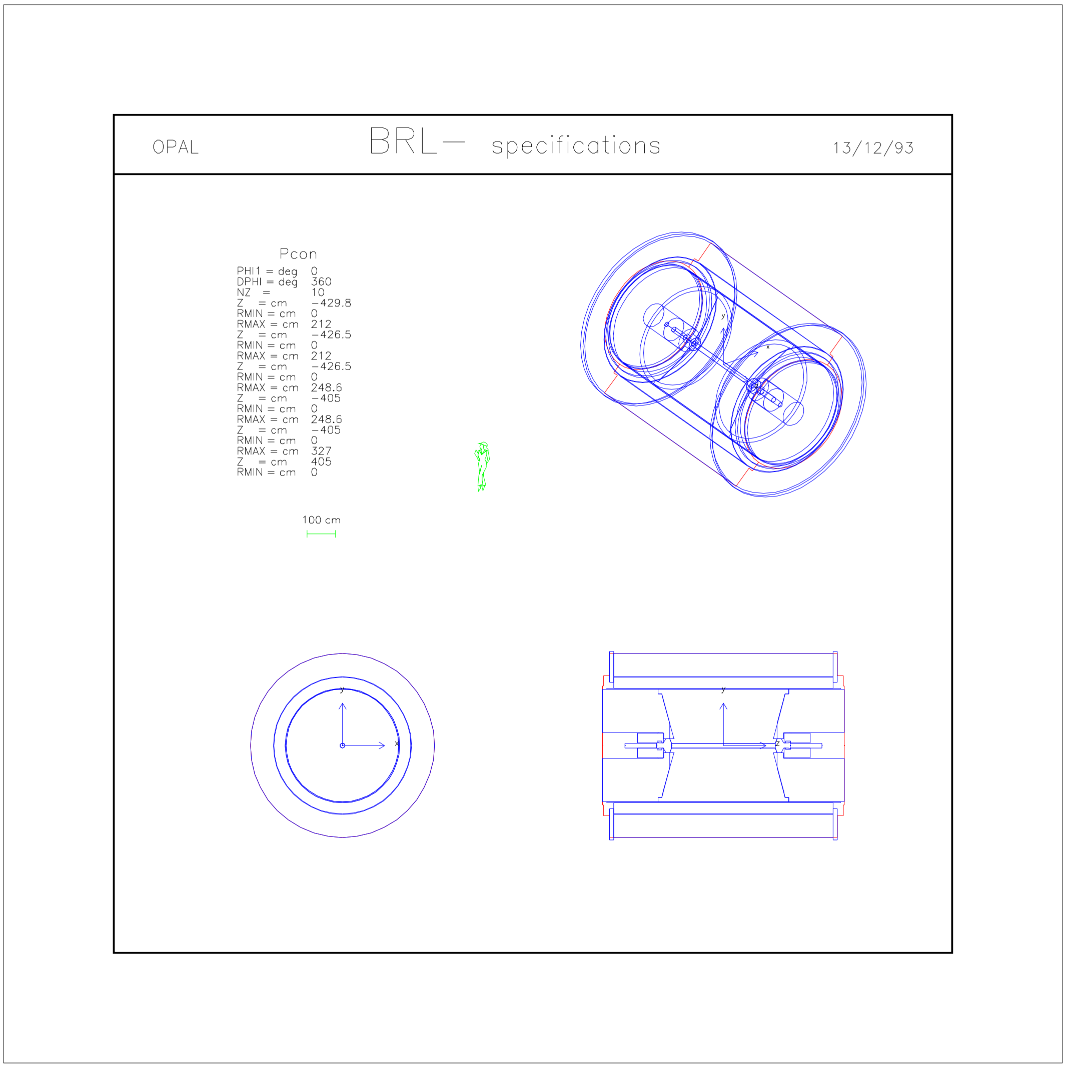


Figure 15: Example of use of GDSPEC

The following information on the volume are presented in a single frame:

- the volume daughters (one level down in the geometrical tree);
- a space view of the volume (with θ=45 and ϕ=135);
- a front view cut;
- a side view cut;
- the axes of the local coordinate system;
- a human figure;
- the scale;
- the shape type;
- all the numerical parameters that define the volume.

In drawing the volumes GDSPEC turns on the sets visibility ('SEEN') attribute for the volume CHNAME itself and its direct descendents. The setting of drawing options (HIDE, CVOL, FILL...) will be respected, allowing to customise the drawing. An example is shown in Fig. 16.

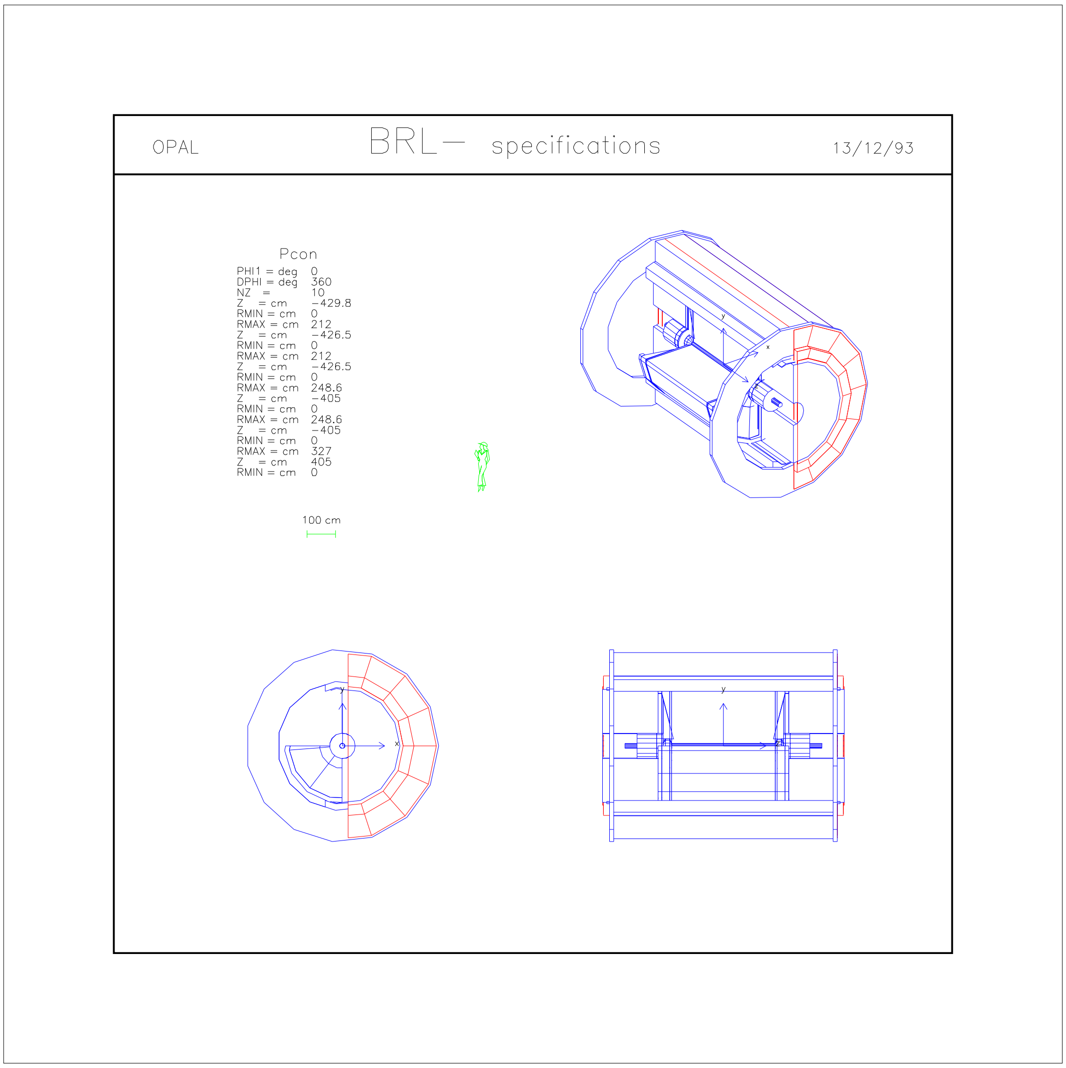


Figure 16: Example of use of GDSPEC

```
CALL GDFSPC (CHNAME,ISORT,INTER)
```

Draws on separate pictures the full set of GDSPEC for the geometrical tree starting from CHNAME, i.e. calls GDSPEC for the volume CHNAME and for all its descendants.

CHNAME (CHARACTER*4) volume name;

ISORT (INTEGER) alphabetic sorting flag;

- $= 1$ all the volumes will be drawn in ascending alphabetic order according to their name;
- $\neq 1$ the volumes will be drawn in the order in which they have been created;

INTER (INTEGER) interactive/batch version flag;

- $= 1$ the routine will prompt the user at each plot before doing a clear screen;
- $\neq 0$ the routine will clear automatically the screen before starting a new frame.

Note: INTER should be set to 1 when using the interactive version of GEANT and to any other value when using a batch version.

Origin : R.Brun, P.Zanarini **Submitted:** 01.10.83
Revision : S.Giani **Revised:** 13.12.93
Documentation : P.Zanarini, S.Giani, F.Carminati

Handling View banks

1 The view banks

The basic detector drawing routines (`GDRAW`, `GDRAWC`, `GDRAWX`) scan the data structure `JVOLUM` repeatedly to extract position and dimension information on the volumes. This can be a rather lengthy process even on fast computer. Hidden line removal drawing requires an auxiliary data structure and the relative visibility of all the volumes has to be analysed. With complicated detectors the time spent in this process can be substantial, depending on the drawing options chosen and on the speed of the machine on which the program is run. In order to alleviate this problem the graphic *banks* and the associated routines have been developed.

The interpretation of the structure (for instance scanning of the volumes' data structure to convert the 3D geometry structure into a set of 2D lines, visibility analysis, surface creation or light processing) is thus separate 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 [DRAW399]`). These views can then be displayed quickly, having only to draw the 2D vectors or the fill areas previously stored.

For a detector with more than 100 different volumes, for example, this costs only a few thousand words of memory for each drawing stored.

If the drawing routines are called while a view bank, identified by a positive integer, is open (`GDOPEN`) only interpretation will be made, no output will be generated until the bank is closed (`GDCLOS`). When a view bank has been closed it cannot be modified, but it can be displayed as many times as wanted (`GDSHOW`) or deleted (`GDELET`).

Once a drawing (detectors, tracks or hits) is in a view bank, it is possible to scan it in detail via the `LENS` or `ZOOM` interactive commands (only available with `X11`). For more information see the `[XINT]` section.

`CALL` **GDOPEN** `(IVIEW)`

`IVIEW` (`INTEGER`) view bank number (`IVIEW>0`).

Opens a *view bank*, used to store 2D graphics information coming from the interpretation of 3D structures (but also 2D annotation). All subsequent calls to the drawing routines will fill the view bank `IVIEW`, without generating any output.

`CALL` **GDCLOS**

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

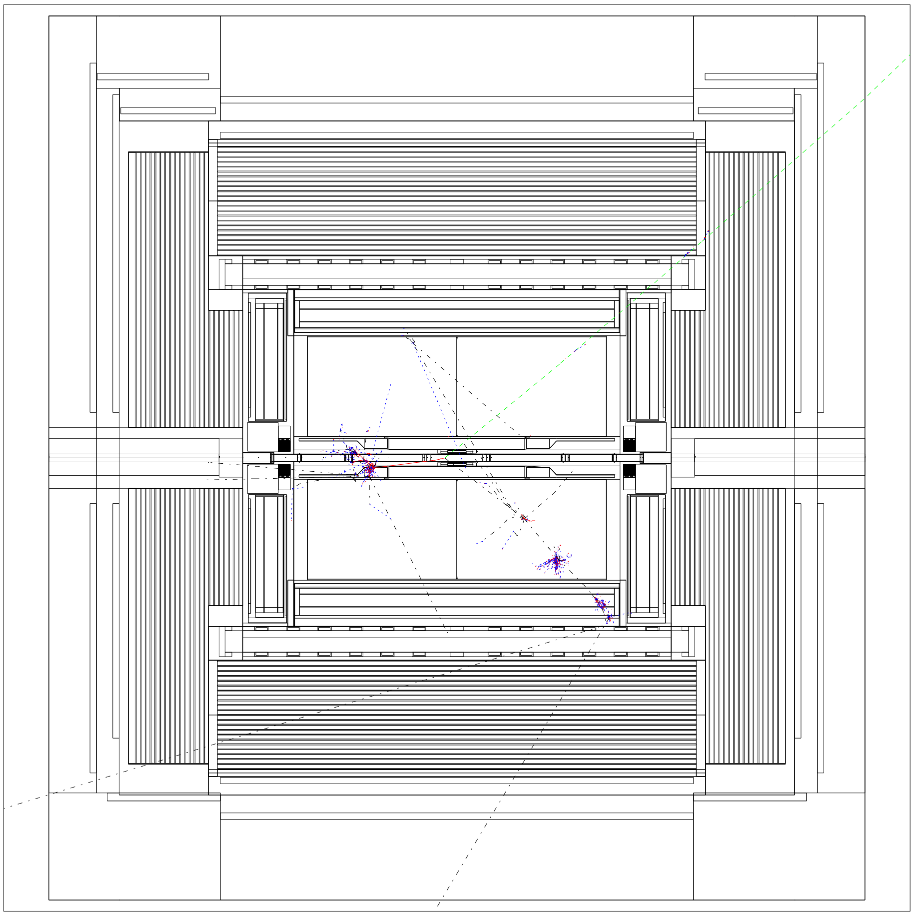
`CALL` **GDSHOW** `(IVIEW)`

`IVIEW` (`INTEGER`) view bank number (`IVIEW>0`).

Displays view bank `IVIEW`. `GDSHOW` can be called either before or after a view bank has been closed. If a view bank is currently open, the content of bank `IVIEW` will be added to it.

`CALL` **GDELET** `(IVIEW)`

`IVIEW` (`INTEGER`) view bank number (`IVIEW>0`).



```
CALL GDOPEN(3)
CALL GDRAWC('ALEF',2,5.,10.,10.,0.013,0.013)
CALL GDCLOS
CALL GDSHOW(3)
.
.
CALL GDXYZ(0)
```

Figure 17: Example of use of view banks

Deletes a view bank from memory. If called before a bank has been closed, GDELET will also restore the screen mode.

Note: the JDRAW data structure has a top level bank which contains one word for each view bank in Q(JDRAW+IVIEW). This word is a key which determines the status of the bank. If this key is 3, the bank itself is protected against deletion, and it cannot be deleted by GDELET. An example of use of view banks is shown in Fig. 17.

Origin	:	R.Brun, P.Zanarini	**Submitted:** 15.11 83
Revision	:		**Revised:** 13.12.93
Documentation	:		

The data structure JDRAW

This data structure contains the so-called *view banks*. The layout of the data structure can be found in Fig. 18. The meaning of the variables is the following:

NKVIEW	number of views stored in the structure;
IVIEW	current view selected;
IGU	current graphic unit pointer;
MAXGU	number of units in graphic unit bank;
MORGU	number of words to push the graphic unit bank;
IGS	current graphic segment pointer;
MAXGS	number of segments in graphic segment bank;
MORGS	number of words to push the graphic segment bank;
ITU	current text unit pointer;
MAXTU	number of units in text unit bank;
MORTU	number of words to push the text unit bank;
ITS	current text segment pointer;
MAXTS	number of segments in text segment bank;
MORTS	number of words to push in text segment bank
LENGU	array of lengths for each graphic unit and of line attributes (LINATT);
ADDGU	array of addresses for each graphic unit;
ADDTU	array of addresses for each text unit;
X	array of u coordinates of graphic segments;
Y	array v coordinates of graphic segments;
ICUT	cut axis (1, 2, 3 or 0 if no cut) of the view;
LINWID	text line width and text attributes (ITXATT);

GTHETA, GPHI, GPSI, GUO, GVO, GSCU, GSCV, are the viewing parameters stored in /GCDRAW/.

UO, VO, SIZE, ANGLE, IOPT, ITEXT have the same meaning of those given as arguments to GDRAWT (or HPLSOF [7]).

A control word is stored in Q(JDRAW+IVIEW), with the following meaning:

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

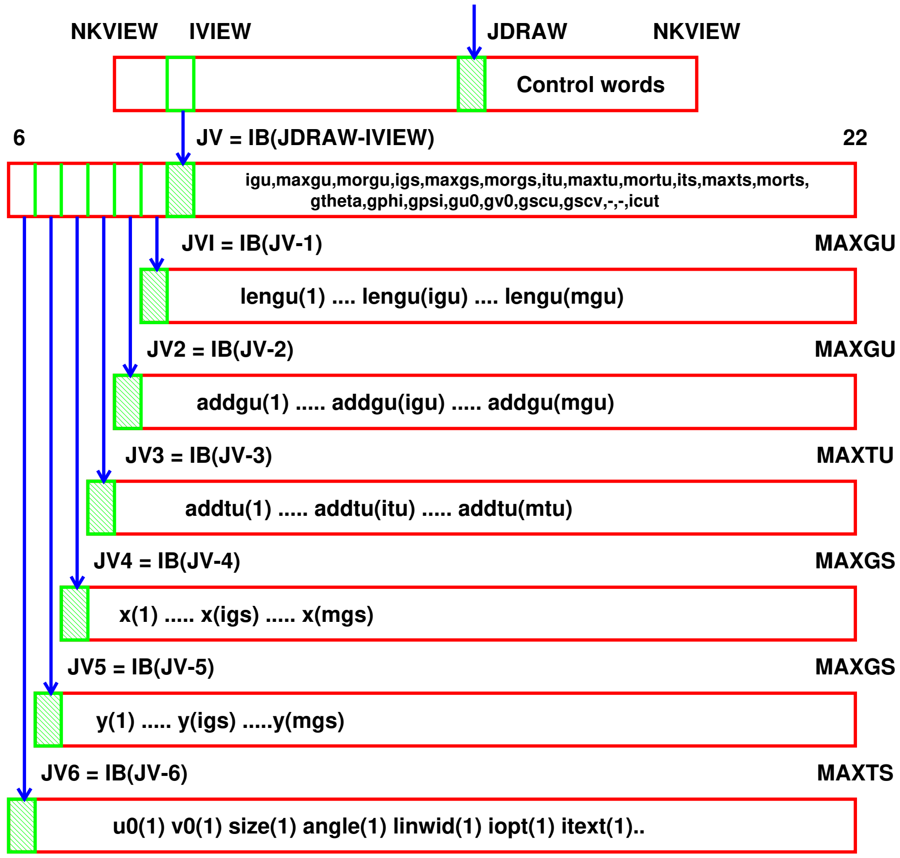


Figure 18: The JDRAW data structure

Origin : P.Zanarini
Revision : S.Giani
Documentation : P.Zanarini, S.Giani, F. Carminati

Submitted: 15.08.83
Revised: 16.12.93

Utility routines of the drawing package

CALL **GDOPT** (CHOPT,CHVAL)

CHOPT (CHARACTER*4) option to be set;
CHVAL (CHARACTER*4) value to be assigned to the option.

Drawing options can be set via the routine GDOPT (corresponding to the DOPT interactive command). The possible value of CHOPT and of CHVAL are the following:

PROJ selects the type of projection, it can have the following values:

- PARA orthographic parallel projection (the default);
- PERS perspective projection.

THRZ acronym for Tracks and Hits in a Radius versus Z projection, where R is the *vertical* axis of the drawing and Z the horizontal one. If a *cut view* has been drawn with the following parameters:

- X-Z cut CUTTHE=90 and CUTPHI=90 in GDRAWX or icut=2 in GDRAWC;
- Y-Z cut CUTTHE=90 and CUTPHI=180 in GDRAWX or icut=1 in GDRAWC.

the different values of the option THRZ have the following effect on the drawing of hits and tracks:

- 360 hits and tracks will be rotated around Z onto the Z-R, R>0 half plane;
- 180 (equivalent to ON) hits and tracks will be rotated around Z onto the Z-R plane by the smaller ϕ angle, so to conserve the sign of their R-ordinate (where R is either X or Y;
- OFF (the default) tracks and hits will be projected from their position in space onto the plane R-Z.

Note: when the THRZ option is set to ON or 180, particles originating in one half space and crossing the horizontal axis will be reflected back into the original half space. If the option 360 can be chosen, all tracks and hits are on the top (positive) half space.

TRAK selects the way tracks are drawn. Possible values:

- LINE (the default) the lines joining track points are drawn by GDXYZ;
- POIN only the track points are drawn.

HIDE selects the hidden line removal technique for the drawings. Possible values are:

- OFF (the default) hidden lines are not removed from the drawing: the drawing is usually rather fast, but the result can be quite messy if a large number of volumes is visible;
- ON hidden lines are not drawn: if this value is selected, the detector can be exploded (interactive command BOMB), cut with different shapes (interactive command menu CVOL), and some parts can be shifted from their original position (interactive command SHIFT). Drawing with hidden line removal can be demanding in terms of CPU time and memory. At the end of each drawing the program displays the memory used and the statistics on the volume visibility. When the drawing requires more memory than available, the program will evaluate and display the number of missing words, so that the user can increase accordingly the size of the ZEBRA store in the common /GCBANK/.

SHAD when HIDE is ON, selects the shading and filling option for the detector's surfaces that are drawn:

- OFF (the default) no shading of the surfaces will be performed;

ON the volume surfaces will be filled according to the value of their FILL and LWID attribute, set via GSATT:

FILL=1 the surfaces are filled with the current colour assigned to the volume via GSATT, 166 different colours are available;

FILL>1 the surfaces are filled with a shade of the basic colour assigned to the volume according to their angle with respect to the line of illumination, the resolution increases with the value of FILL ($1<$ FILL ≤ 7);

LWID>0 just as the number of scan-lines used to fill a surface depends on the value of FILL, their width depends on the value of LWID: increasing the value of LWID will result in thicker lines.

EDGE control drawing of the volume edges in HIDE=ON, FILL=ON mode:

ON the edges of the faces are not drawn, so that they will be visible only as intersections of filled faces with different luminosity;

OFF the edges of the faces are drawn. For the volumes which are not filled, the normal edges will be drawn irrespective of the value of this option.

CALL **GDZOOM** (ZFU,ZFV,UZO,VZO,U0,V0)

ZFU (REAL) zoom factor for U coordinate;
ZFV (REAL) zoom factor for V coordinate;
UZO (REAL) U coordinate of centre of zoom;
VZO (REAL) V coordinate of centre of zoom;
U0 (REAL) U coordinate of centre of zoom in the resulting picture;
V0 (REAL) V coordinate of centre of zoom in the resulting picture.

This routine sets the zoom parameters (part of the viewing parameters in common /GCDRAW/) that define how objects will be displayed by the subsequent drawing operations. The zoom is computed around UZO, VZO (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.

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

X0 (REAL) X coordinate of origin;
Y0 (REAL) Y coordinate of origin;
Z0 (REAL) Z coordinate of origin;
AXSIZ (REAL) axis size in centimeters.

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

CALL **GDSCAL** (U0,V0)

U0 (REAL) U coordinate of the centre of the scale;
V0 (REAL) V coordinate of the centre of the scale.

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.

CALL **GDMAN** (U0,V0)

CALL **GWOMA1** (U0,V0)

CALL **GDWMN2** (U0,V0)

CALL **GDWMN3** (U0,V0)

UO (REAL) U coordinate of the centre of the figure;
VO (REAL) V coordinate of the centre of the figure.

Draws the profile of a man, or three different female figures in 2D user coordinates, with the current scale factors. The figure is approximately 160 cm high.

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

U (REAL) U position of text string;
V (REAL) V position of text string;
CHTEXT (CHARACTER*(*)) text string;
SIZE (REAL) character size (cm);
ANGLE (REAL) rotation angle (degrees);
LWIRTH (INTEGER) line width (1,2,3,4,5);
IOPT (INTEGER) centering option for the text:

-1	left adjusted;
0	centered;
1	right adjusted.

Draws text with software characters. It has the same arguments of the HPLOT [7] routine HPLSOF (a call to HPLSOF is actually performed).

```
CALL GDRAWV (U,V,NP)
```

U (REAL) array of U coordinates;
DT (REAL) array of V coordinates;
NP (INTEGER) number of points.

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

```
CALL GDHEAD (ISEL,CHNAME,CHRSIZ)
```

ISEL (INTEGER) option to be selected for the title name (decimal integer):

0	only the header lines;
xxxxx1	add the text CHNAME centered on top of header;
xxxx1x	add global detector name (first volume) on left;
xxx1xx	add date on right;
xx1xxx	select thick characters for text in top of head (i.e. with larger line width);
x1xxxx	add the text EVENT NR x on top of header;
1xxxxx	add the text RUN NR x on top of header.

Note: ISEL=x1xxx1 or ISEL=1xxxx1 are illegal choices, i.e. they generate overwritten text.
CHNAME (CHARACTER*4) title string;
CHRSIZ (REAL) character size (cm) of text CHNAME.

This routine draws a frame header.

```
CALL GDCOL (ICOL)
```

This routine has been kept for backward compatibility. The appropriate HIGZ routines should be called instead.

ICOL (INTEGER) colour code (1,2,. . . ,8), it can be positive, negative, or zero:

>0	to set the colour permanently;
<0	to set the colour temporarily;

=0 to restore the permanent colour value.

This routine sets the colour code. 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.

```
CALL GDLW (LW)
```

This routine has been kept for backward compatibility. The appropriate HIGZ routines should be called instead.

LW (INTEGER) line width code (1,2,. . . ,5), it can be positive, negative, or zero:

- >0 to set the line width permanently;
- <0 to set the line width temporarily;
- $=0$ to restore the permanent line width value.

This routine sets the line width. 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.

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

U0 (REAL) U coordinate of the graphics cursor;
V0 (REAL) V coordinate of the graphics cursor;
ICHAR (INTEGER) ASCII value of the key pressed.

A graphics input is provided by this routine to fetch the 2D user coordinates of the graphics cursor on the screen. Interactive commands to zoom, measure, or pick tracks or hits make use of this routine. When the routine is called in the interactive version of GEANT, the present position of the graphics cursor on the screen becomes visible and it can be moved with the mouse. The user then positions the cursor and presses the left button. The routine returns then in U0, V0 the user coordinates of the graphics cursor. On terminals which do not have the mouse, the graphic cursor is moved with the arrows, and the point is acquired when any key, except *carriage return*, is pressed. In this case the routine returns in ICHAR the ASCII value of the key pressed. If the user types the carriage return, the previous value is retained.

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

X (REAL) array X(3,NPOINT) of space points;
NPOINT (INTEGER) number of points:

- >0 X is in the current DRS;
- <0 X is in MARS;

U (REAL) array of NPOINT U coordinates;
V (REAL) array of NPOINT V coordinates.

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

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

θ the angle between the line of sight and the Z axis ($0 \leq \theta \leq 180$);
ϕ the angle between the X axis and the projection onto the X-Y plane of the line of sight ($0 \leq \phi \leq 360$).

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*)
```

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

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 θ and ϕ angles).

DRAW Bibliography

[1] R.Bock et al. *HIGZ Users Guide,* Program Library Q120. CERN, 1991.

[2] ISO. The graphical kernel system (gks). Technical Report ISO 7942, ISO Geneva, 1985.

[3] ISO. The graphical kernel system for three dimensions (gks-3d). Technical Report ISO 8805, ISO Geneva, 1987.

[4] D.R.Myers. *GKS/GKS-3D Primer,* DD/US/110. CERN, 1987.

[5] J.Gettys, R.W.Scheiffer, and R.Newman. 1989.

[6] ISO. Programmer's hierarchichal interactive graphics system (phigs), – part 1: Functional description. Technical Report ISO 9592-1, ISO Geneva, 1989.

[7] R.Brun and H.Renshall. *HPLOT users guide,* Program Library Y251. CERN, 1990.

Origin :
Revision :
Documentation : F.Bruyant

Submitted: 01.10.84
Revised: 10.03.94

The geometry package

The geometry package has two main functions:

1. define, during the initialisation of the program, the geometry in which the particles will be tracked;
2. communicate, during the event processing phase, to the tracking routines the information for the transport of the particles in the geometry which has been defined.

The present section reviews the concepts of the geometry package and explains how the geometrical information should be provided by the user. It is important to point out that, once the geometry has been defined, the tracking of particles through the different volumes proceeds without any intervention from the user [TRAK].

The connection between the geometry and tracking packages is established by the subroutines GMEDIA/GTMEDI, GTNEXT/GNEXT and GINVOL which answer respectively the questions:

- In which volume is a given point ?
- What is the distance to the nearest volume along the trajectory of the particle ?
- What is the distance to the nearest volume ?
- Is a given point still in the current volume ?

The routines GTMEDI, GTNEXT are used in a dynamic context, at tracking time, when the knowledge of the track direction can be used to save time.

1 The volume definition

Experimental setups, as complex as the LEP[1] detectors, can be described rather accurately through the definition of a set of *volumes*. Each *volume* is given a *name* and is characterised by:

- a *shape* identifier, specifying one of the basic geometrical shapes available [GEOM050];
- the *shape* parameters, giving the dimensions of the volume;
- a local reference system, with origin and axes defined for the each shape;
- the physical properties, given by a set of constants describing 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 neighbouring volumes, magnetic field, etc.) [CONS, TRAK];
- a set of attributes, in connection with the drawing package and the detector response package [DRAW, HITS].

Until it is *positioned* in a given reference frame, a volume is an entity which has no spatial relation with the other volumes. By convention, a unique initial volume has to be defined first which will contain all the other volumes. The reference frame intrinsic to this volume is considered to be the master reference frame.

There are sixteen basic shapes in GEANT, which are described in [GEOM050].

[1] The Large Electron Positron collider in operation at CERN.

2 Volumes with contents

A *volume* can be declared to have contents and as such it becomes 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 obtained by specifying its translation and rotation with respect to the mother reference frame. The user should make sure that no volume extends beyond the boundaries of its mother.

When a volume is positioned, the user gives it a *number*. Multiple copies of the same volume can be positioned inside the same or different mothers as long as their copy numbers differ. The contents of the positioned volume are reproduced in all copies.

Mother volumes can be divided by planes along the three axes (x, y, z), radially along R_{xy}, or along the spherical coordinates of a polar system: R, θ and ϕ. The axis along which it is possible to perform a division depend on the shape to be divided. The general rule is that the result of the division be still a `GEANT` shape.

A volume can be partially or totally divided. The division process generates a number of *cells*, which are considered as new volumes. The cell dimensions are computed according to the number of divisions or the step size. A cell, as any other volume, can again be divided, or have other volumes positioned into it. Any operation of positioning or division performed on one cell is repeated to all the cells resulting from the division. A volume can be *either* divided *or* have contents, but not the two things together.

These operations define a physical tree with several levels. The material and properties of the contents replace the ones of the mother within the space region they occupy. A `volume` is therefore defined not only by its intrinsic characteristics but also by those of its *descendants*, namely its contents (by division or positioning), the contents of its contents, etc.

3 Overlapping volumes, see also `[GEOM020]`

The user may define volumes which have no relation with the real objects. It is sometimes convenient to make use of such volumes, to artificially delimit regions with simple shapes. Doing this, it may happen that volumes partially overlap each other. (A volume positioned inside a mother is obviously not regarded as overlapping the mother).

It is possible in `GEANT` to handle partially overlapping volumes, but there are some implications that the user should be aware of. A flag `'ONLY'`/`'MANY'` is assigned to each copy of a given volume at the moment this is positioned. The `'MANY'` option indicates that a point found in this volume could also be in other volumes which are not its direct descendants. If one of the overlapping volumes where the point is found is declared as `'ONLY'`, the geometry subroutines will give priority to this volume, and the search stops. If a point is inside several `'MANY'` volumes and outside all `'ONLY'` ones, priority will be given to the volume found at the deepest level, or to the first found, if there are many valid `'MANY'` candidates at the same level. In order to avoid ambiguities, two overlapping `'MANY'` volumes should in general have the same tracking medium.

4 The physical tree

An example of a 4 level physical tree with embedded volumes and the corresponding geometrical configuration are shown in fig 19.

where A and B are `BOX`es, C a `TRAP`ezoid, and D and E `TUBE`s (see `[GEOM050]` for more detail). 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 the fine details of a complex structure.

5 The data structures `JVOLUM` and `JGPAR` and the common `/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

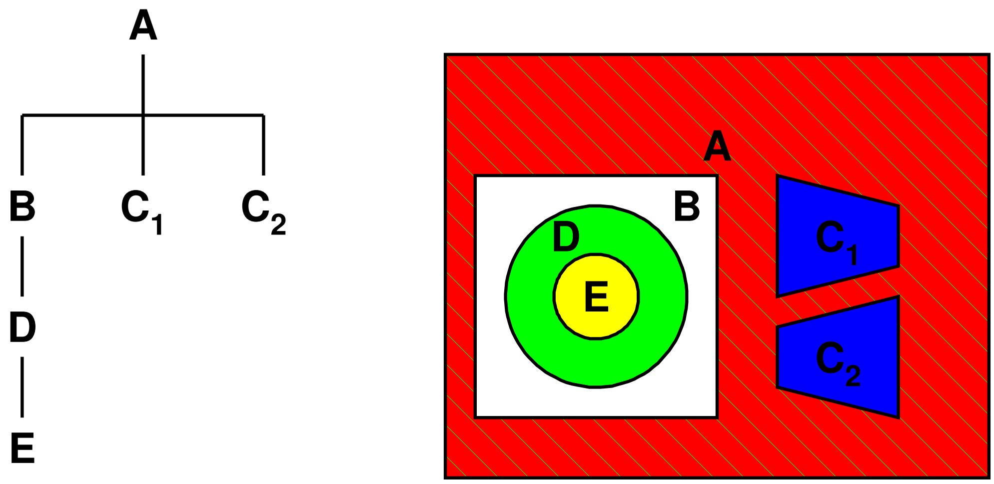


Figure 19: Example of geometrical tree structure

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 volume in the logical tree and a region in space. Information has to be added 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`/`GTMEDI`, for the current point of the current track, in the common `/GCVOLU/`. It includes the current level number `NLEVEL` and, for each level, starting from the first mother volume, the identification of the corresponding volume, e.g.: `NAME`, `NUMBER`, `'ONLY'`/`'MANY'` flag, translation and rotation with respect to the master reference frame and so on. The number of shape parameters and pointers to the vector describing the actual values of those parameters, for each level, are stored in the data part and in the link part, of the data structure `JGPAR`, respectively.

6 The user tools

The geometry through which the particles are transported is defined by the user via a set of calls to subroutines of the `GEANT` package.

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

`GSVOLU` define (*instantiate*) a volume by giving a `NAME` and the actual parameters to a shape; the position of the volume inside the bank `JVOLUM` is returned. If any of the parameters which express a length is negative, `GEANT` will assign to this parameter at tracking time the maximum value allowed, without protruding out of the mother which contains the particle transported.
A shape can be instantiated into a volume also without actual parameters. These will then be supplied when positioning each copy of the volume via the `GSPOSP` routine (see below). This allows to have many *similar* volumes with the same name and shape, but different actual parameters in each copy.

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

`GSPOS` position a copy of a volume inside its mother with respect to the mother's reference system. A a point in space, a rotation matrix index, a copy number and the `'ONLY'`/`'MANY'` flag are supplied;

GSPOSP position a copy of a volume which has been instantiated without actual parameters inside its mother with respect to the mother reference system. The parameters of the copy being positioned are supplied as well. The volumes will be identified by name and copy number as for multiple copies of the same volume.

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

GSDVN divide a volume in a given *number* of cells completely filling the mother. In this case the cell tracking medium is assumed to be the same as for the mother. See also GSDVN2.

GSDVT divide a volume with slices of a given *step*. The cell tracking medium can be different from the one of the mother. See also GSDVT2.

GSDVX divide a volume starting from a given offset. In addition to 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 must be specified.

7 Geometrical information retrieval

The parameters of a volume may depend on the physical tree which leads to this volume. The dimensions of one of the slices of a cone divided along Z depends on which slice we consider. So a volume is completely defined only if we specify the path in the tree. This is composed by the name and copy number of the volumes containing the one we are interested in, from the first mother.

Given this information, the routine GLVOLU is capable to fill the common /GCVOLU/, and the structure JGPAR with the actual parameters of the instance chosen.

Origin :
Revision :
Documentation : S.Giani, S.Ravndal

Submitted: 10.03.94
Revised: 10.03.94

Tracking inside volumes and optimisation

The tracking of particles through the geometrical data structure is the key functionality of GEANT. At every particle's step, the program must find the volume where the particle is (GTMEDI) and the next boundary it will cross (GTNEXT). This can take about 60% of the total simulation time (even for detectors described in an optimized way); therefore, a new logic has been introduced to minimize the time spent for the search in the geometrical tree.

1 Virtual divisions

Instead of a linear or binary search (time spent proportional or logarithmic with the number of volumes, respectively), a 'direct access' technique has been developed to make the time basically independent from the number of volumes. Every volume containing other volumes is 'virtually' divided in equal slices at initialization time via the call to GGCLOS (the best axis is computed automatically). For each slice, a data structure is filled with a list of the volumes identifiers intersecting such slice. Slices with identical lists are pointing to the same contents and are collected. At tracking time it is immediate to find in which slice the particle is and only its contents have to be checked. The same is true to find the next boundary to be crossed: only if the intersection point with a content lies outside the current collection of slices, the next one will be checked. The algorithm gives in average about a factor 2 in speed for the overall simulation in the large detectors. It also minimizes the dependency of the performance on the skill and experience of the programmer and allows a fast tracking even in geometrical structures received from CAD systems.

2 Other optimisation tools

The following facilities are kept for backward compatibility reasons. If called by the user, these optimisation tools are called at initialisation time, but not used at tracking time. Instead, the new optimisation using 'virtual divisions' is performed. If the user wants to use the older optimisation tools, he can recompile GEANT 3.21 using the PATCHY flag +OLD.

1. GSORD/GGORD

   From the position of the contents inside a given volume, the subroutine GSORD computes fictitious boundaries along the specified coordinate, simulating a division with irregular 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 overlapping with that pseudo-cell, as sketched in fig 20.

   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/GSNEAR

   when a particle exits from a daugther of a mother volume, the contents are scanned initially in the order in which 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 subroutines GSNEXT/GSNEAR permit the user to inject, at initialisation time, for each content in turn, the list of neighbours to search for. GSNEAR is recommended, GSNEXT is kept for backward, compatibility and has a different default option. A proper use of this facility can reduce the search time significantly.

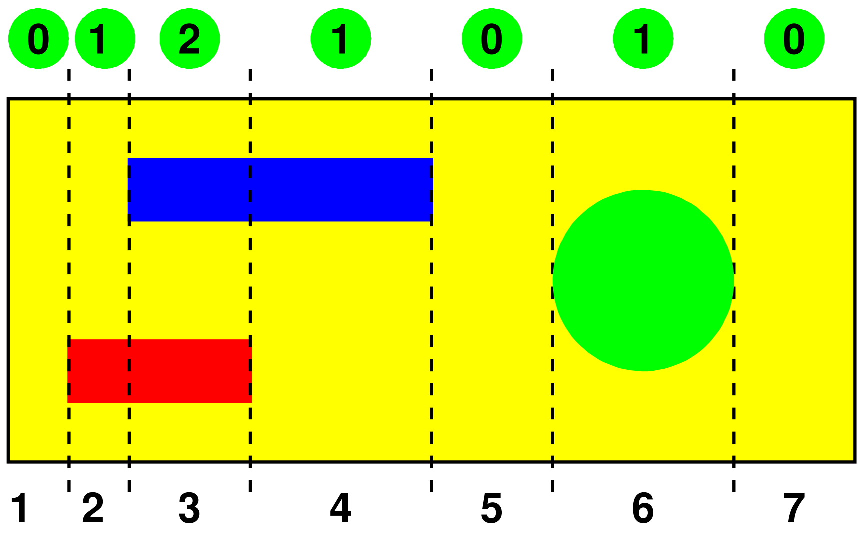


Figure 20: Example of pseudo-cells created by GSORD

3. GSUNEA/GUNEAR

   The volumes to be checked when exiting from one daughter can depend on the direction and position of the exiting particles. To optimise tracking taking into account these dynamic features, the user can code the GUNEAR routine, and inform GEANT that it has to use it at tracking time by calling the GSUNEA routine.

Origin :
Revision :
Documentation : S.Giani, S.Ravndal

Submitted: 22.03.94
Revised: 22.03.94

'MANY' Volumes and boolean operations on volumes

The topology of GEANT is given by 16 general shapes which are used to represent solids in the design of detectors. These shapes are bounded by 2-nd order surfaces for reasons of speed at tracking time. New tools have been introduced to extend the power of the GEANT geometry: boolean operations (union, intersection and subtraction) between such shapes are now allowed, so that an infinity of new shapes can be built; divisions along arbitrary axis are now possible, allowing to exploit any simmetry in the design. The tracking is now able to handle overlapping volumes in the most general sense: the concept of MANY volumes has been extended to automatic clipping for protuding objects. Thanks to this MANY technique, any 3D or 2D geometrical pattern can be reduced to a single dimensional structure, basically cancelling the need for GUNEAR at tracking time. Finally, more than one order of magnitude in speed has been gained compared to the old step search algorithm used to track in MANY volumes.

- Topological extensions via boolean operations.
- Divisions along arbitrary axis.
- Reduction of 2D structure to 1D structure.
- Logic and algorithm.

1 Topological extensions via boolean operations

Boolean operations between a given set of shapes allow the creation of an infinite number of shapes. Adding a single new primitive (for ex. a toroid) to the basic set of shapes, all the possible combinations with all the old shapes become moreover possible.

- UNION: the union of two volumes B and C is obtained positioning two overlapping 'MANY' daughters B and C inside the same mother A. To identify the result of the union as a single volume is enough to create a SET associated to the volumes B and C.
- SUBTRACTION: the subtraction of a volume B from a volume C is obtained positioning B as 'ONLY' and C as a 'MANY' overlapping B. The result of the subtraction is automatically what the tracking sees as the C volume, so GSPOS is the only user interface needed. The volume B can have the same material as the mother A or not. The normal positioning technique used for 'ONLYs' is a particular case of boolean subtraction (a volume is contained inside the other).
- INTERSECTION: the intersection of two volumes B and C is obtained positioning a protuding 'MANY' object C in a 'MANY' object B which is normally placed in a 'ONLY' mother A. A and B must have the same material, while C has the material asked for the intersection. The intersection is given by the part of C which is not protuding from B, therefore by what the new tracking sees as the volume C: again the only needed user interface is GSPOS. If not interacting with other daughters of A, B could also be 'ONLY'. Without any ambiguity, even further 'MANY' daughters of A can overlap the protuding part of C outside B, because it is really invisible to the new tracking.

2 Divisions along arbitrary axis

As protuding `MANY` daughters are automatically clipped by their mother, it is possible to divide objects along arbitrary directions. Suppose for example that we want to divide a PCON into planar slices parallel to the plane X + Y = 0; it is enough to position a `MANY` BOX embedding completely the PCON 'inside' the PCON itself; the BOX must be rotated of 45 degrees in X and Y and then divided along one of its natural axis. Another example can be the following: suppose we want to divide 24 times in PHI a PGON with only 6 natural segmentations in PHI; then we have to position a `MANY` TUBE, completely embedding the PGON, 'inside' the PGON itself; again, we can now divide the tube in 24 slices in PHI and the tracking will see only the PGON divided 24 times in PHI.

3 Reduction of 2D structure to 1D structure

Sometimes there are complicated multidimensional geometrical structures (like honeycombs, spaghetti, etc.) which can be reduced to the overlap of several one-dimensional patterns (for which divisions can be used or virtual divisions are particularly efficient). The `MANY` option allows to superimpose such 1D structures to reproduce the multi-dimensional ones in a very effective way, instead of positioning the volumes one by one in the 2D or 3D pattern. A picture will follow which explains this concept better.

4 Logic and algorithm

In `GTMEDI`, all the `'MANY'` volumes for which the point is found inside (and not in any of its daughters) is put in a stack saving also its branch of the tree. The point is considered to be in the deepest level `'MANY'`. Each time the point is also found in an intermediate `'ONLY'` volume the stack is cleared (to insure boolean subtraction). In `GTNEXT`, if stepping inside a `'MANY'` volume, not `ONLY` its daughters and its own boundaries are checked, but also the daughters and the boundaries of all the others `'MANY'` in the list have to be checked. This is not yet enough: the logic requires that the boundaries of all the possible overlapping volumes are checked, so it is also necessary for each `'MANY'` volume of the list to go up to the first `'ONLY'` mother in the tree repeating the same procedure. This allows also boolean intersection. Obviously, volumes already checked are not checked again.

Origin : R.Brun, A.McPherson
Revision :
Documentation :

Submitted: 15.08.83
Revised: 08.11.93

The GEANT shapes

The GEANT geometry package offers sixteen basic shapes with which to describe the setups where particles are transported. In this section we will describe these shapes. For each shape figs (21), (22), (23), (24) show a simple drawing illustrating the meaning of the parameters and the position and orientation of the reference frame proper of that shape. A description of the shapes and of their parameters follows. Angles are always in degrees. With every shape is given the code which is used internally by GEANT to identify it.

1 BOX box with faces perpendicular to the axes. It has 3 parameters:

1. DX half-length of the box along the x-axis;
2. DY half-length of the box along the y-axis;
3. DZ half-length of the box along the z-axis;

2 TRD1 trapezoid with the x dimension varying along z. It has 4 parameters:

1. DX1 half-length along x at the z surface positioned at -DZ;
2. DX2 half-length along x at the z surface positioned at +DZ;
3. DY half-length along the y-axis;
4. DZ half-length along the z-axis;

3 TRD2 trapezoid with both x and y dimensions varying along z. It has 5 parameters:

1. DX1 half-length along x at the z surface positioned at -DZ;
2. DX2 half-length along x at the z surface positioned at +DZ;
3. DY1 half-length along y at the z surface positioned at -DZ;
4. DY2 half-length along y at the z surface positioned at +DZ;
5. DZ half-length along the z-axis;

4 TRAP *general* trapezoid: the faces perpendicular to z are trapezia and their centres are not necessarily on a line parallel to the z axis. This shape has 11 parameters, but only considering that the faces should be planar, only 9 are really independent. A check is performed on the user parameters and a message is printed in case of non-planar faces. Ignoring this warning may cause unpredictable effect at tracking time.

1. DZ half-length along the z axis;
2. THET polar angle of the line joining the centre of the face at -DZ to the centre of the one at +DZ;
3. PHI azimuthal angle of the line joining the centre of the face at -DZ to the centre of the one at +DZ;
4. H1 half-length along y of the face at -DZ;
5. BL1 half-length along x of the side at -H1 in y of the face at -DZ in z;
6. TL1 half-length along x of the side at +H1 in y of the face at -DZ in z;
7. ALP1 angle with respect to the y axis from the centre of the side at -H1 in y to the centre of the side at +H1 in y of the face at -DZ in z;
8. H2 half-length along y of the face at +DZ;

9. BL2 half-length along x of the side at -H2 in y of the face at +DZ in z;
10. TL2 half-length along x of the side at +H2 in y of the face at +DZ in z;
11. ALP2 angle with respect to the y axis from the centre of the side at -H2 in y to the centre of the side at +H2 in y of the face at +DZ in z;

5 TUBE tube. It has 3 parameters:

1. RMIN inside radius;
2. RMAX outside radius;
3. DZ half length in z;

6 TUBS ϕ segment of a tube. It has 5 parameters:

1. RMIN inside radius;
2. RMAX outside radius;
3. DZ half length in z;
4. PHI1 starting angle of the segment;
5. PHI2 ending angle of the segment;

PHI1 should be smaller than PHI2. If this is not the case, the system adds 360 degrees to PHI2.

7 CONE conical tube. It has 5 parameters:

1. DZ half-length in z;
2. RMN1 inside radius at -DZ in z;
3. RMX1 outside radius at -DZ in z;
4. RMN2 inside radius at +DZ in z;
5. RMX2 outside radius at +DZ in z;

8 CONS ϕ segment of a conical tube. It has 7 parameters:

1. DZ half-length in z;
2. RMN1 inside radius at -DZ in z;
3. RMX1 outside radius at -DZ in z;
4. RMN2 inside radius at +DZ in z;
5. RMX2 outside radius at +DZ in z;
6. PHI1 starting angle of the segment;
7. PHI2 ending angle of the segment;

PHI1 should be smaller than PHI2. If this is not the case, the system adds 360 degrees to PHI2.

9 SPHE segment of spherical shell. It has 6 parameters:

1. RMIN inside radius of the shell;
2. RMAX outside radius of the shell;
3. THE1 starting polar angle of the shell;
4. THE2 ending polar angle of the shell;
5. PHI1 starting azimuthal angle of the shell;
6. PHI2 ending azimuthal angle of the shell;

10 PARA parallelepiped. It has 6 parameters:

1. DX half-length in x;
2. DY half-length in y;
3. DZ half-length in z;

4.`ALPH` angle formed by the y axis and by the plane joining the centre of the faces parallel to the z-x plane at `-DY` and `+DY`;

5.`THET` polar angle of the line joining the centres of the faces at `-DZ` and `+DZ` in z;

6.`PHI` azimuthal angle of the line joining the centres of the faces at `-DZ` and `+DZ` in z;

`11 PGON` *polygon*. It has at least 10 parameters:

1.`PHI1` the azimuthal angle ϕ at which the volume *begins* (angles are counted counterclockwise);

2.`DPHI` *opening* angle of the volume, which extends from `PHI1` to `PHI1+DPHI`;

3.`NPDV` number of sides of the cross section between the given ϕ limits;

4.`NZ` number of planes perpendicular to the z axis where the dimension of the section is given – this number should be at least 2 and `NP` triplets of numbers must follow;

5.`Z` z coordinate of the section;

6.`RMIN` radius of the circle tangent to the sides of the inner polygon in the cross-section;

7.`RMAX` radius of the circle tangent to the sides of the outer polygon in the cross-section;

`12 PCON` *polycone*. It has at least 9 parameters:

1.`PHI1` the azimuthal angle ϕ at which the volume *begins* (angles are counted counterclockwise);

2.`DPHI` *opening* angle of the volume, which extends from `PHI1` to `PHI1+DPHI`;

3.`NZ` number of planes perpendicular to the z axis where the dimension of the section is given – this number should be at least 2 and `NP` triplets of numbers must follow;

4.`Z` z coordinate of the section;

5.`RMIN` radius of the inner circle in the cross-section;

6.`RMAX` radius of the outer circle in the cross-section;

`13 ELTU` elliptical cross-section tube. It has three parameters:

1.`P1` semi-axis of the ellipse along x;

2.`P2` semi-axis of the ellipse along y;

3.`DZ` half-length in z;

The equation of the surface is $x^2\mathtt{P1}^{-2} + y^2\mathtt{P2}^{-2} = 1$.

`14 HYPE` hyperbolic tube, i.e. the inner and outer surfaces are hyperboloids, as would be formed by a system of cylindrical wires which were then rotated tangentially about their centres. It has 4 parameters:

1.`RMIN` inner radius at z=0, where tube is narrowest;

2.`RMAX` outer radius at z=0, where tube is narrowest;

3.`DZ` half-length in z;

4.`THET` *stereo angle* of rotation of the two faces;

The hyperbolic surfaces are given by: $r^2 = (z\tan\theta)^2 + r^2_{z=0}$

`28 GTRA` general twisted trapezoid: the faces perpendicular to z are trapezia and their centres are not necessarily on a line parallel to the z axis as the `TRAP`; additionally, the faces may be *twisted* so that none of their edges are parallel. It is a `TRAP` shape, except that it is *twisted* in the x-y plane as a function of z. The parallel sides perpendicular to the z axis are rotated with respect 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 trapezoidal face at z=-DZ to the corresponding points on the face at z=DZ. Divisions are not allowed. It has 12 parameters:

1. DZ half-length along the z axis;
2. THET polar angle of the line joining the centre of the face at -DZ to the centre of the one at +DZ;
3. PHI azimuthal angle of the line joining the centre of the face at -DZ to the centre of the one at +DZ;
4. TWIST *twist* angle of the faces parallel to the x-y plane at z = ±DZ around an axis parallel to z passing through their centre;
5. H1 half-length along y of the face at -DZ;
6. BL1 half-length along x of the side at -H1 in y of the face at -DZ in z;
7. TL1 half-length along x of the side at +H1 in y of the face at -DZ in z;
8. ALP1 angle with respect to the y axis from the centre of the side at -H1 in y to the centre of the side at +H1 in y of the face at -DZ in z;
9. H2 half-length along y of the face at +DZ;
10. BL2 half-length along x of the side at -H2 in y of the face at +DZ in z;
11. TL2 half-length along x of the side at +H2 in y of the face at +DZ in z;
12. ALP2 angle with respect to the y axis from the centre of the side at -H2 in y to the centre of the side at +H2 in y of the face at +DZ in z;

Note: this shape suffers from the same limitations than the TRAP: the tracking routines assume that the faces are planar, but this constraint is not easily expressed in terms of the 12 parameters. Additionally, no check on the faces is performed in this case. Users should avoid to use this shape as much as possible, and if they have to do so, they should make sure that the faces are really planes. If this is not the case, the result of the transport is unpredictable.
To accelerate the computations necessary for transport, 18 additional parameters are calculated for this shape:

1. DXODZ dx/dz of the line joining the centres of the faces at z=±DZ;
2. DYODZ dy/dz of the line joining the centres of the faces at z=±DZ;
3. XO1 x at z=0 for line joining the + on parallel side, perpendicular corners at z=±DZ;
4. YO1 y at z=0 for line joining the + on parallel side, + on perpendicular corners at z=±DZ;
5. DXDZ1 dx/dz for line joining the + on parallel side, + on perpendicular corners at z=±DZ;
6. DYDZ1 dy/dz for line joining the + on parallel side, + on perpendicular corners at z=±DZ;
7. XO2 x at z=0 for line joining the - on parallel side, + on perpendicular corners at z=±DZ;
8. YO2 y at z=0 for line joining the - on parallel side, + on perpendicular corners at z=±DZ;
9. DXDZ2 dx/dz for line joining the - on parallel side, + on perpendicular corners at z=±DZ;
10. DYDZ2 dy/dz for line joining the - on parallel side, + on perpendicular corners at z=±DZ;
11. XO3 x at z=0 for line joining the - on parallel side, - on perpendicular corners at z=±DZ;
12. YO3 y at z=0 for line joining the - on parallel side, - on perpendicular corners at z=±DZ;
13. DXDZ3 dx/dz for line joining the - on parallel side, - on perpendicular corners at z=±DZ;
14. DYDZ3 dy/dz for line joining the - on parallel side, - on perpendicular corners at z=±DZ;
15. XO4 x at z=0 for line joining the + on parallel side, - on perpendicular corners at z=±DZ;
16. YO4 y at z=0 for line joining the + on parallel side, - on perpendicular corners at z=±DZ;
17. DXDZ4 dx/dz for line joining the + on parallel side, - on perpendicular corners at z=±DZ;
18. DYDZ4 dy/dz for line joining the + on parallel side, - on perpendicular corners at z=±DZ;

29 CTUB *cut* tube, a tube cut at the extremities with planes not necessarily perpendicular to the z axis. It has 11 parameters:

1. `RMIN` inside radius;
2. `RMAX` outside radius;
3. `DZ` half length in z;
4. `PHI1` starting angle of the segment;
5. `PHI2` ending angle of the segment;
6. `LX` x component of a unit vector perpendicular to the face at `-DZ`;
7. `LY` y component of a unit vector perpendicular to the face at `-DZ`;
8. `LZ` z component of a unit vector perpendicular to the face at `-DZ`;
9. `HX` x component of a unit vector perpendicular to the face at `+DZ`;
10. `HY` y component of a unit vector perpendicular to the face at `+DZ`;
11. `HZ` z component of a unit vector perpendicular to the face at `+DZ`;

`PHI1` should be smaller than `PHI2`. If this is not the case, the system adds 360 degrees to `PHI2`.

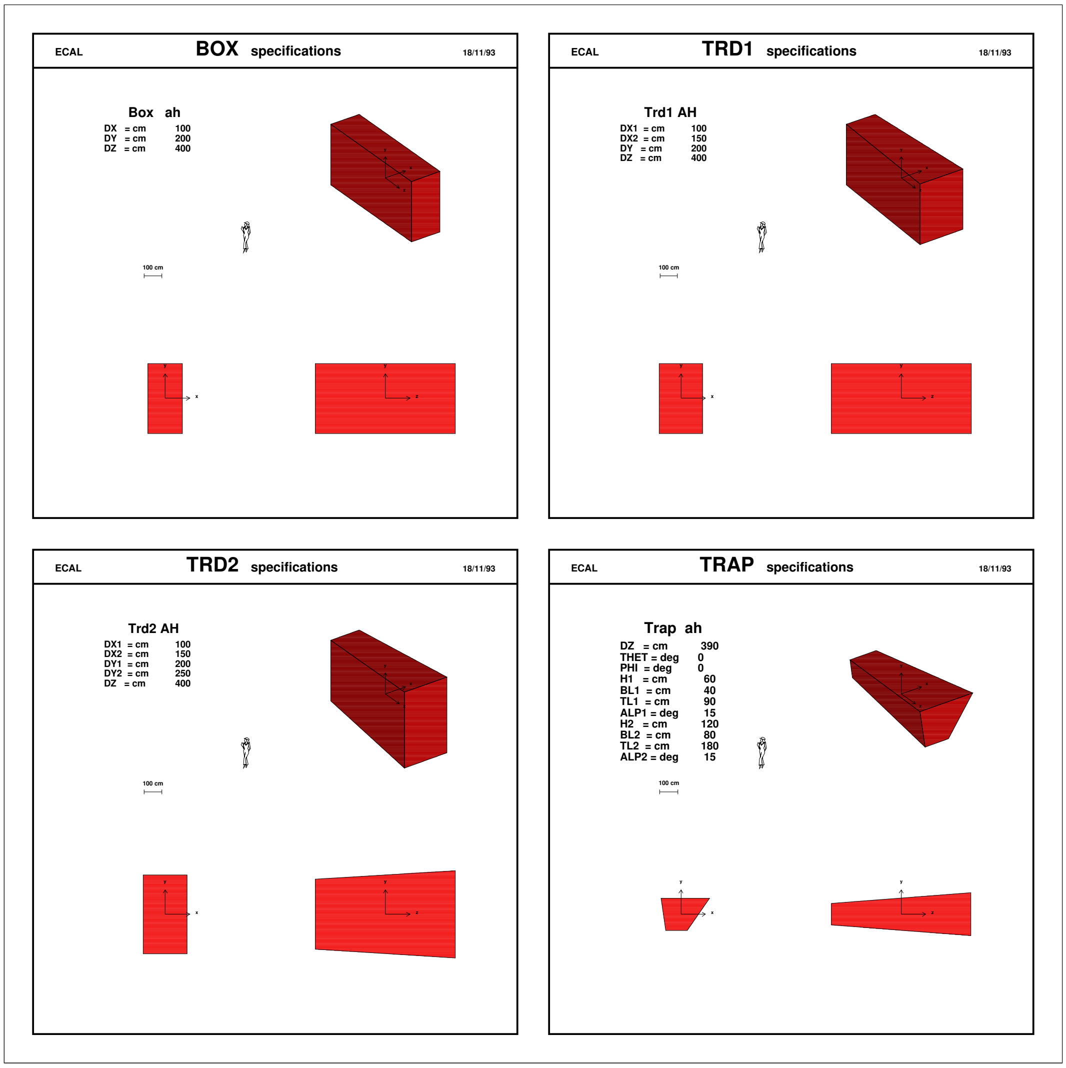


Figure 21: shapes `BOX`, `TRD1`, `TRD2`, `TRAP`

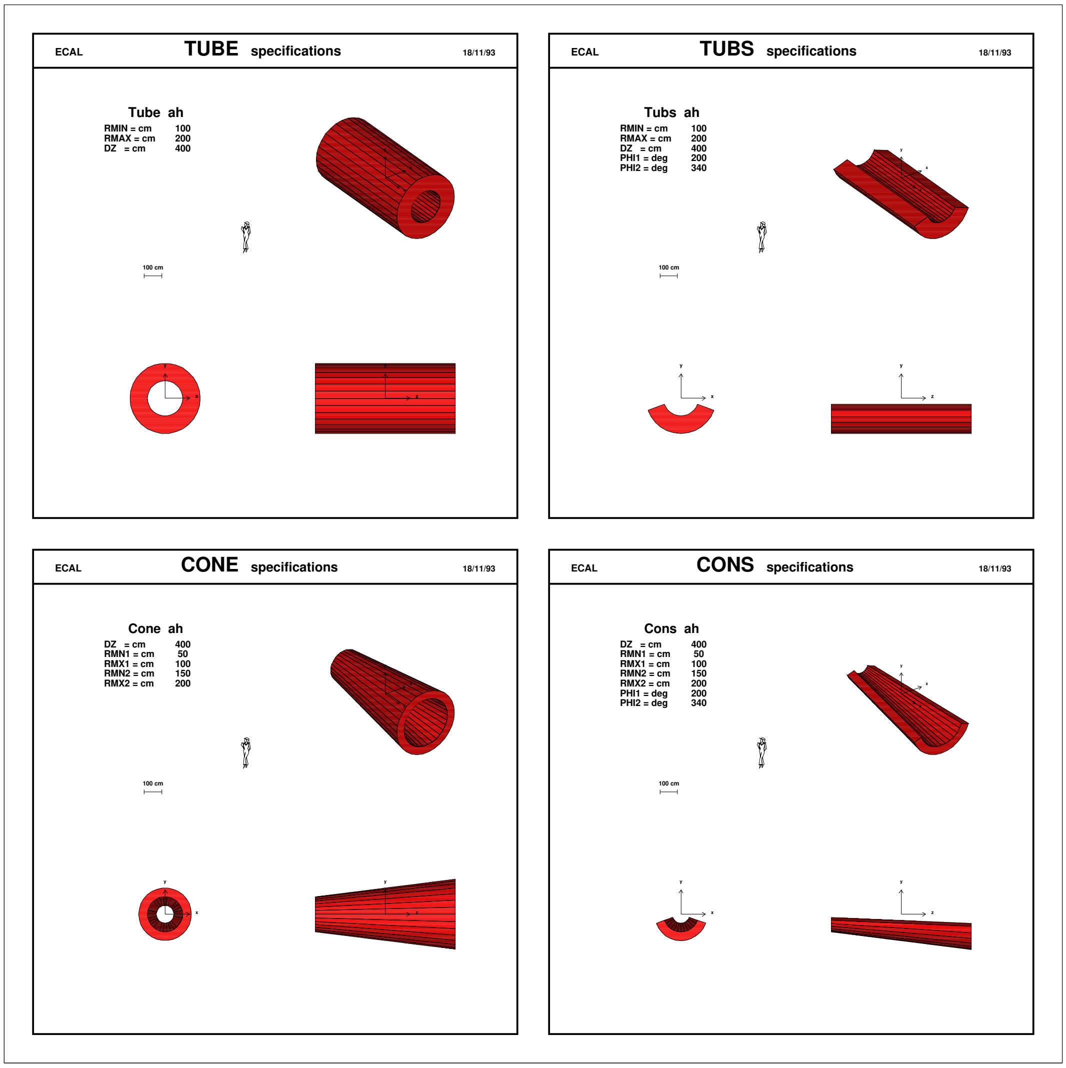


Figure 22: shapes `TUBE`, `TUBS`, `CONE`, `CONS`

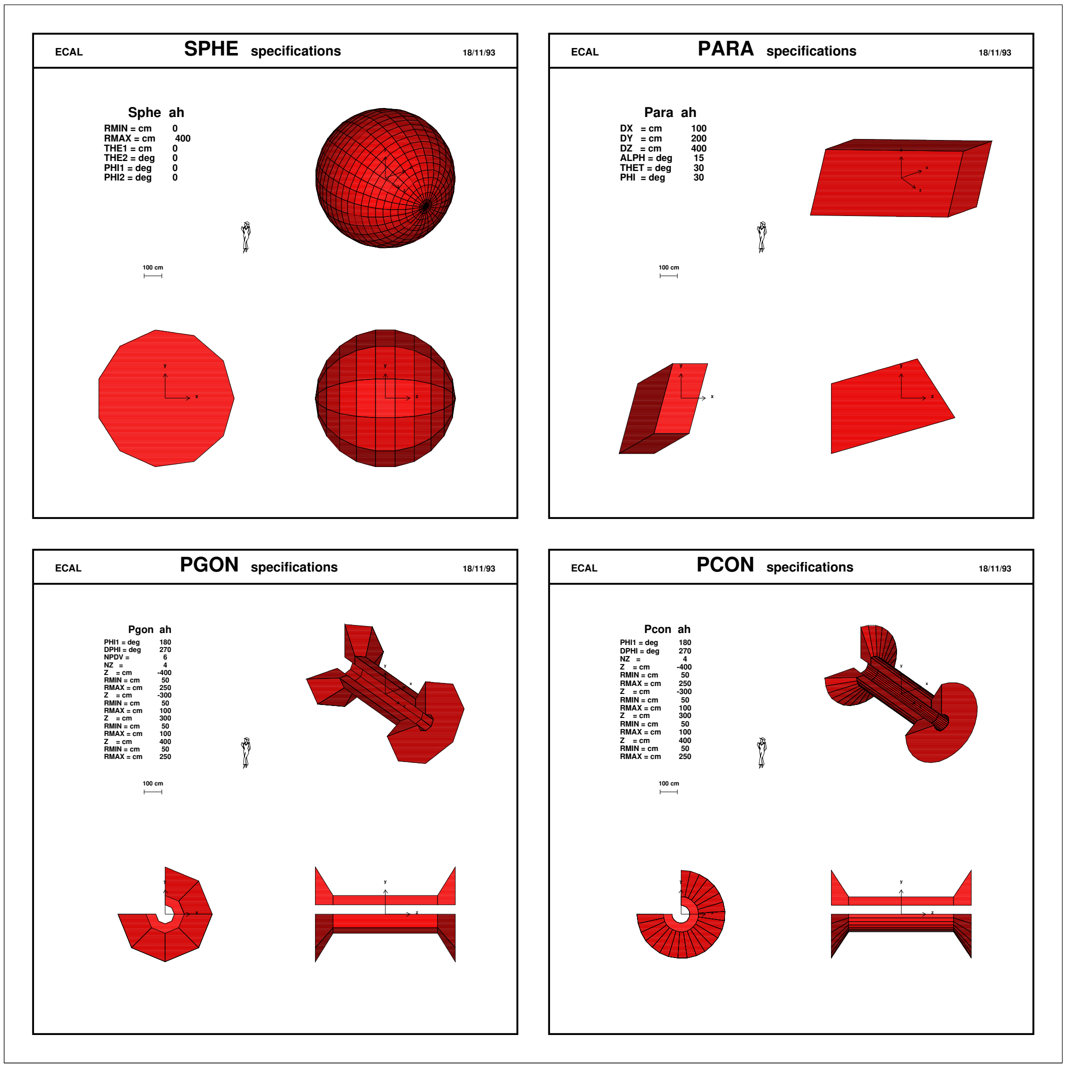


Figure 23: shapes `PARA`, `SPHE`, `PGON`, `PCON`

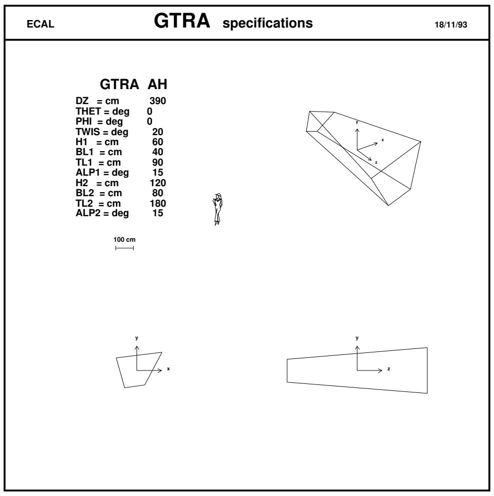


Figure 24: shapes GTRA

Origin : R.Brun,A.McPherson **Submitted:** 15.08.83
Revision : **Revised:** 16.11.93
Documentation :

Creation of a volume

CALL **GSVOLU** (CHNAME,CHSHAP,NMED,PAR,NPAR,IVOLU*)

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

CHNAME	(CHARACTER*4) volume name – it must be unique;
CHSHAP	(CHARACTER*4) name of one of the GEANT shapes;
NMED	(INTEGER) tracking medium number for the volume – if other volumes are positioned within this one, their tracking medium replaces the one of the mother;
PAR	(REAL) array containing the shape parameters;
NPAR	(INTEGER) number of parameters – if zero, then the volume must be positioned via the routine GSPOSP indicating the parameters for each copy (see [GEOM120]);
IVOLU	(INTEGER) internal volume number – if ≤ 0 an error condition has happened.

If one of the parameters expressing a length is negative, GEANT will try to maximise its value in each copy, without extending beyond the limits of the mother. This facility can be used in conjunction with the division of volumes using GSDVN, GSDVT, and GSDVX. Restriction apply to this facility:

- the daughter should be positioned without a rotation matrix;
- all shapes can be positioned within a similar shape with negative parameters, but only in the centre of the mother;
- boxes with negative parameters can be positioned in: box, TRD1, TRD2 and TRAP with $\theta = \phi = 0$;
- HYPE, ELTU and CTUB cannot have negative parameters;
- PCON and PGON can only have 2 z planes.

Examples

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

CALL GSVOLU('ECAL','BOX ',1,PECAL,3,IVOLU)
```

CALL **GPVOLU** (IVOLU)

Prints the volume parameters:

IVOLU (INTEGER) GEANT number of the volume to print – if IVOLU ≤ 0, all volumes will be printed;

Origin : R.Brun, A.McPherson
Revision :
Documentation :

Submitted: 15.08.83
Revised: 16.11.93

Positioning a volume inside its mother

```
CALL GSPOS (CHNAME,NR,CHMOTH,X,Y,Z,IROT,CHONLY)
```

Places a copy of a volume previously defined by a call to `GSVOLU` inside its mother volume `CHMOTH`.

`CHNAME`	(`CHARACTER*4`) name of the volume being positioned;
`NR`	(`INTEGER`) copy number of the volume `CHNAME` being positioned;
`CHMOTH`	(`CHARACTER*4`) name of the volume in which copy `NR` of `CHNAME` is positioned;
`X`	(`REAL`) x position of the volume in the mother reference system;
`Y`	(`REAL`) y position of the volume in the mother reference system;
`Z`	(`REAL`) z position of the volume in the mother reference system;
`IROT`	(`INTEGER`) rotation matrix number describing the orientation of the volume relative to the coordinate system of the mother;
`CHONLY`	(`CHARACTER*4`) flag to indicate whether a point found to be in this volume may also be in other volumes which are not direct descendants of it – possible values are `ONLY` and `MANY`.

The following points are important for a correct use of `GSPOS`:

- the position and rotation with which a volume is positioned are relative to the mother reference system, that is to the reference system of the volume which contains the copy being positioned. To see how the reference system is defined for the `GEANT` shapes, see [GEOM050];
- `GSPOS` can be called several times with the same name, to place multiple copies of the same volume either in the same mother volume or in different ones. The copy number parameter `NR` is user defined, and it allows the different copies of the same volume to be distinguished. The user is free to assign any valid integer to this parameter. If two volumes are positioned with the same copy number, it will be impossible to decide in which one a particle is during tracking. If the same volume is positioned twice inside the same mother with the same copy number, the parameters of the second call to `GSPOS` will override the first ones for that copy;
- all the copies of a volume are identical: in particular they contain all the same daughters recursively – conversely, positioning a volume inside a mother will insert it in all the copies of that mother;
- the volumes mentioned in the call must be already defined by a call to `GSVOLU`;
- no volume can be positioned in a divided mother;
- in case of positioning without rotation, `IROT` should be set to 0 – note that this is faster than defining a unit matrix;
- the first volume, containing all the apparatus, cannot be positioned;
- if the `CHONLY` flag is equal to `ONLY`, the volume being positioned should not overlap with any other volume except his mother and his daughters and should not extend beyond the limits of its mother. In this case the search routines assume that, when a point is found in this volume, the only further searching required is in its contents. Other volumes at the same or higher levels or in separate branches will not be looked at. The first volume must be `ONLY`;

- if the CHONLY flag is set to MANY, a point found in this volume can also be in another volume at the same or higher level or in a different branch of the tree. To decide where a point is the following algorithm is applied:

  1. the tree is scanned *upward* until an ONLY mother with *positioned* contents (i.e. non-divided) is found;
  2. all the branches of the tree descending from this volume are scanned to find other volumes where the point may be;
  3. a winner is selected among the candidates according to the following rules:
     - if one of the candidates is ONLY, the point is declared in this volume. If more than one candidate is ONLY there is an error in the geometry definition;
     - amongst several MANY candidates, the one at the deepest level is chosen;
     - amongst several MANY candidates at the same level, the first one found is chosen;

  Although a considerable effort has been put in making sure that the MANY option works properly and caters for all the possible cases, still it introduces a considerable complication in the tracking and performance can be rather poor.

  Users should resort to this only when it is not possible to describe the experimental setup with ONLY volumes.

The data structure is so designed that, in any case, the parameters of a volume and the information about the number and position of its daughters are stored only once. This allows the creation of very complex geometries without correspondingly large storage requirements.

Origin : R.Brun, F.Bruyant, M.Maire, A.McPherson
Revision :
Documentation :

Submitted: 15.08.83
Revised: 18.11.93

Positioning a volume inside its mother with parameters

```
CALL GSPOSP (CHNAME,NR,CHMOTH,X,Y,Z,IROT,CHONLY,PAR,NPAR)
```

Position a copy of volume `CHNAME` inside its mother `CHMOTH` assigning its parameters.

`CHNAME`	(`CHARACTER*4`) name of the volume being positioned;
`NR`	(`INTEGER`) copy number of the volume `CHNAME` being positioned;
`CHMOTH`	(`CHARACTER*4`) name of the volume in which copy `NR` of `CHNAME` is positioned;
`X`	(`REAL`) x position of the volume in the mother reference system;
`Y`	(`REAL`) y position of the volume in the mother reference system;
`Z`	(`REAL`) z position of the volume in the mother reference system;
`IROT`	(`INTEGER`) rotation matrix number describing the orientation of the volume relative to the coordinate system of the mother;
`CHONLY`	(`CHARACTER*4`) flag to indicate whether a point found to be in this volume may also be in other volumes which are not direct descendants of it – possible values are `ONLY` and `MANY`;
`PAR`	(`REAL`) array of parameters for the volume being positioned;
`NPAR`	(`INTEGER`) number of parameters.

It is often the case in a detector to have a family of similar objects, differing only by their dimensions. A typical example are the crystals of an electromagnetic calorimeter. In this case it is convenient and logically consistent to give them all the same name, but assign to each copy a different set of parameters.

This can be done defining the volume with 0 parameters through the routine `GSVOLU` and then assigning the parameters to each copy via the routine `GSPOSP`.

`GSPOSP` is similar in all other aspects to `GSPOS` and the user is referred to the description of this routine for more information. An obvious limitation is that a volume defined with 0 parameters can only be positioned via `GSPOSP`, because otherwise its dimensions will be undefined.

Origin : R.Brun, A.McPherson
Revision :
Documentation :

Submitted: 29.09.83
Revised: 18.11.93

Division of a volume into a given number of cells

```
CALL GSDVN (CHNAME,CHMOTH,NDIV,IAXIS)
```

Divide a volume in a given number of parts along a direction.

- `CHNAME` (`CHARACTER*4`) a unique name for the volume to be generated by subdivision of the mother volume;
- `CHMOTH` (`CHARACTER*4`) volume that has to be subdivided;
- `NDIV` (`INTEGER`) number of divisions into which the mother volume is to be divided;
- `IAXIS` (`INTEGER`) *axis* of the division.

The volumes resulting from a division are true volumes in the `GEANT` sense, with the only limitation that they cannot be positioned. They can have daughters and they can still be divided in turn. Different divisions are distinguished by a different division number, much the same as different copy of the same volume by the copy number. Every operation performed on a division cell (positioning or further division), is automatically propagated to all divisions cells of the same mother.

The kind of the division is indicated by the `IAXIS` argument. The allowed values of `IAXIS` and the resulting divisions are shown in figs. 25, 26, 27, 28, where the pictures in the first column correspond to an `IAXIS` value equal to 1, the second column to `IAXIS` = 2, and the third column to `IAXIS` = 3 respectivly. The basic constraint imposed by the `GEANT` geometry is that the result of a division is a volume which can be described in terms of the basic shapes.

The local coordinate system of the divisions is parallel to the system of the mother in case of divisions obtained by parallel planes or concentric divisions. In the case of parallel axes the coordinate system is shifted so that it is in the centre of the division. In case of ϕ divisions produced by planes passing through the axis of a volume with axial symmetry (`TUBE, TUBS, CONE, CONS, PGON, PCON, SPHE`), the local coordinate systems will be rotated so that the x axis passes through the centre of the division.

This routine allows the user to generate a large number of identical volumes filling a defined volume. The material of the slices is inherited from the mother.

It may happen that the different divisions of a volume have different dimensions. In this case, if a volume has to be positioned in each division, its dimensions must be different according to the division number. This can be obtained in some special cases giving to the dimension of the volume which has to vary in order to fit into the different mothers, the value of -1 in the call to `GSVOLU` which defines it. As explained in `[GEOM050]`, `GEANT` will take care to assign to this dimension the maximum value in order to fill the current mother.

What can be done with division can be done also, in most cases, explicitly positioning all the copies. Users must be aware, however, that divisions offer the best performance in tracking of all `GEANT` geometrical constructs, and they should be used instead of positioning whenever possible in the definition of objects.

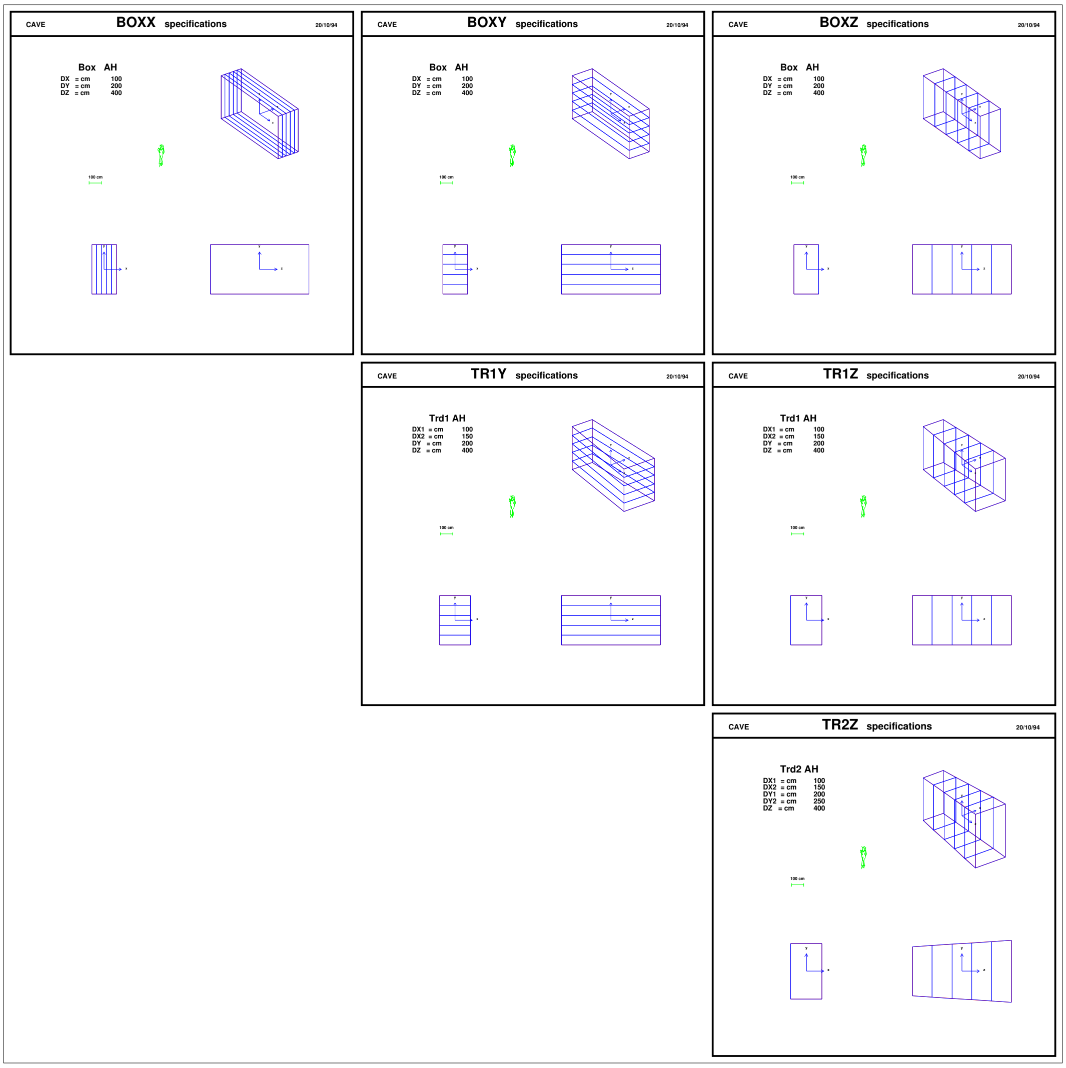


Figure 25: shapes `BOX`, `TRD1`, `TRD2`

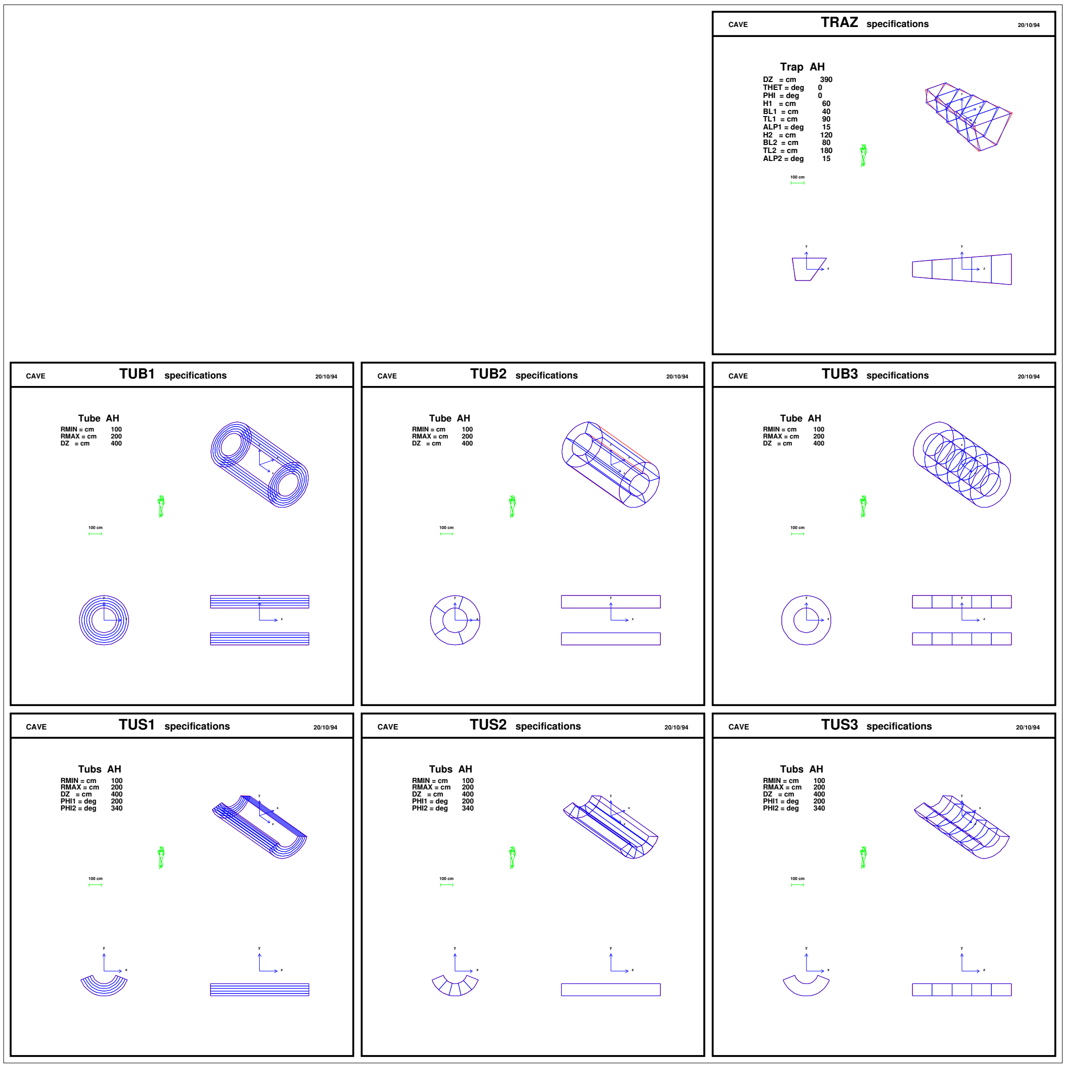


Figure 26: shapes `TRAP`, `TUBE`, `TUBS`

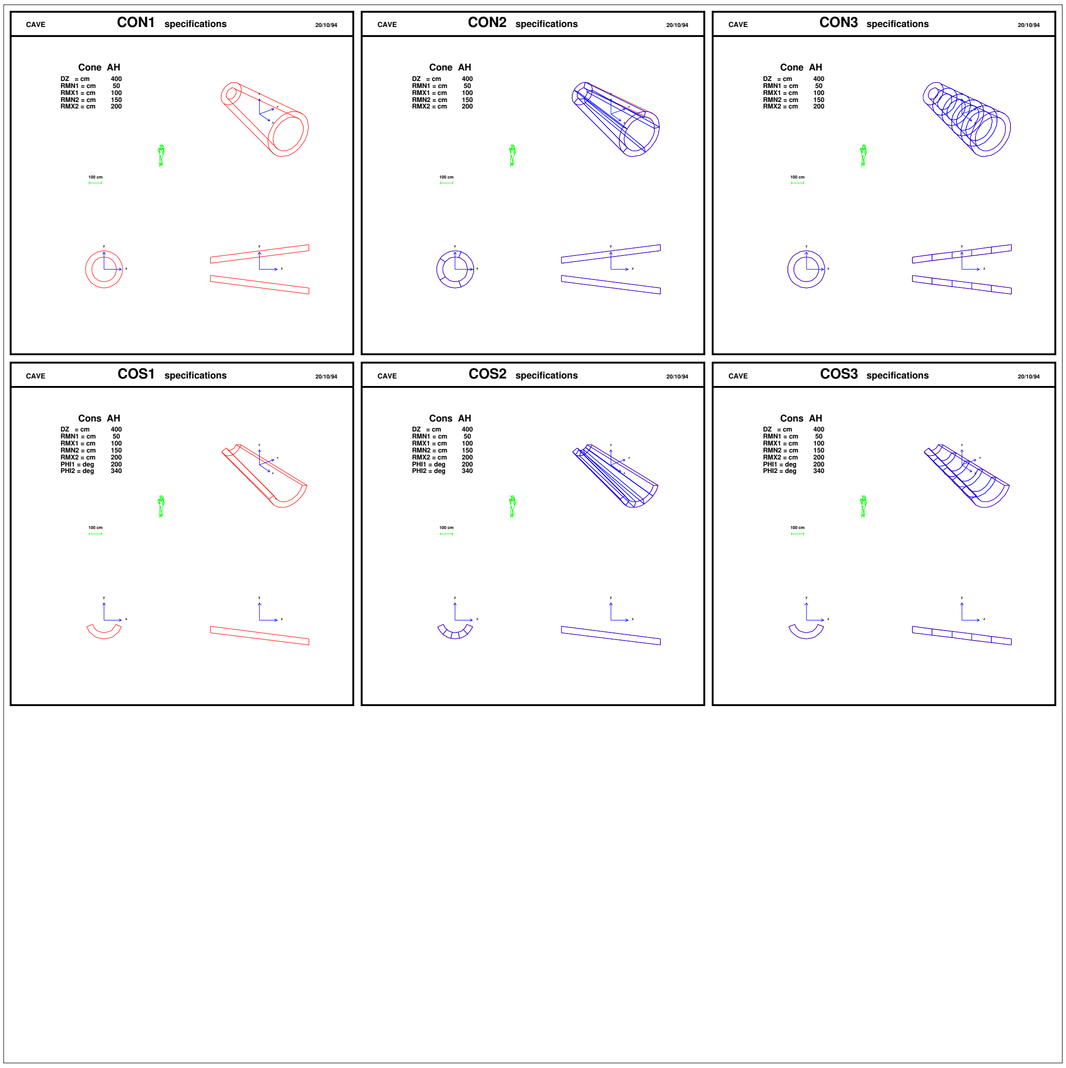


Figure 27: shapes `CONE, CONS`

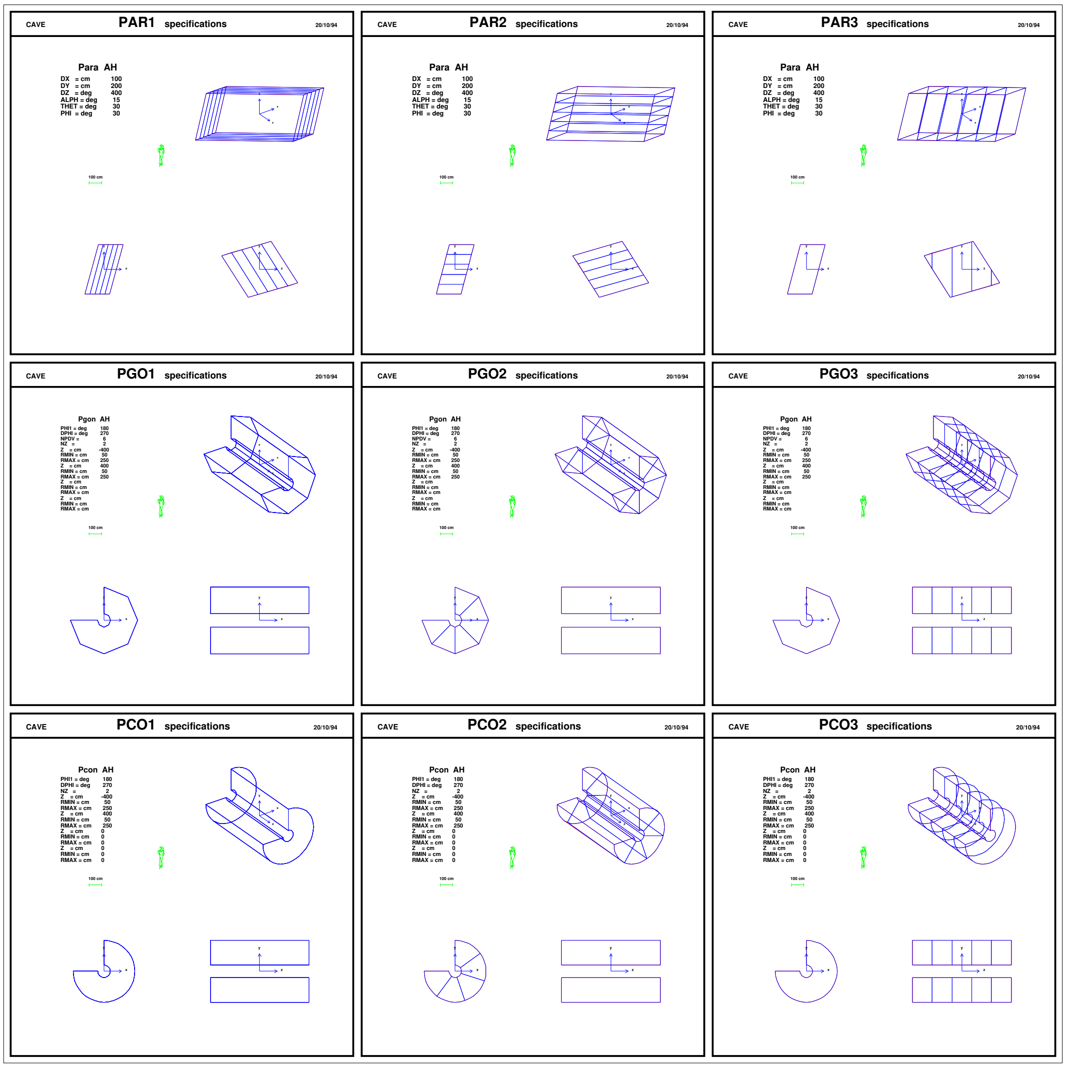


Figure 28: shapes `PARA`, `PGON`, `PCON`

Origin : F.Bruyant, A.McPherson, M.Maire
Revision :
Documentation :

Submitted: 17.12.83
Revised: 18.11.93

Division of a Volume into cells of a given size

This routine creates new volumes from a mother by divisions of a given step.

```
CALL GSDVT (CHNAME,CHMOTH,STEP,IAXIS,NUMED,NDVMX)
```

- `CHNAME` (`CHARACTER*4`) a unique name for the volume to be generated by subdivision of the mother volume;
- `CHMOTH` (`CHARACTER*4`) volume that has to be subdivided;
- `STEP` (`REAL`) size of the divisions – this value can be in centimeters or degrees according to the value of `IAXIS`;
- `IAXIS` (`INTEGER`) *axis* of the division.
- `NUMED` (`INTEGER`) medium number of the divisions – this can be different from the one of the mother, as the division cells may leave a portion of the mother undivided (see below) – if `NUMED` ≤ 0 the medium of the mother;
- `NDVMX` (`INTEGER`) expected (maximum) number of divisions – if ≤ 0 or > 255, 255 is assumed.

The full range of the mother will be divided in sections of the user supplied step. If the step is such that the mother cannot be divided exactly, the largest possible number of divisions will be generated, the excess space will be equally divided between each end of the range of the mother. These extra spaces will be assumed to belong to the mother.

For more information on the division in general, see [GEOM130].

Origin : R.Brun, F.Bruyant, A.McPherson
Submitted: 17.12.83
Revision :
Revised: 18.11.93
Documentation :

Division of a volume - general case

```
CALL GSDVX (CHNAME,CHMOTH,NDIV,IAXIS,STEP,CO,NUMED,NDVMAX)
```

Divide a volume in a given number of parts along a direction, with a given step starting from an offset.

- `CHNAME` (`CHARACTER*4`) a unique name for the volume to be generated by subdivision of the mother volume;
- `CHMOTH` (`CHARACTER*4`) volume that has to be subdivided;
- `NDIV` (`INTEGER`) number of divisions into which the mother volume is to be divided;
- `IAXIS` (`INTEGER`) *axis* of the division.
- `STEP` (`REAL`) size of the divisions – this value can be in centimeters or degrees according to the value of `IAXIS`;
- `CO` (`REAL`) offset where division should start – this value can be in centimeters or degrees according to the value of `IAXIS`;
- `NUMED` (`INTEGER`) medium number of the divisions – this can be different from the one of the mother, as the division cells may leave a portion of the mother undivided (see below) – if `NUMED` ≤ 0 the medium of the mother;
- `NDVMX` (`INTEGER`) expected (maximum) number of divisions – if ≤ 0 or > 255, 255 is assumed.

For more information on the division mechanism, see `[GEOM130]` and `[GEOM140]`. For the moment either `NDIV` or `STEP` must be set negative or 0, so that they will be computed from the `CHMOTH`'s size. The case with both `NDIV` and `STEP` positive is not coded yet. It would permit leaving different gaps at both ends of the `CHMOTH`.

Provisionally the code consists of a call to either `GSDVN2` or `GSDVT2`.

```
CALL GSDVN2 (CHNAME,CHMOTH,NDIV,IAXIS,CO,NUMED)
```

Divide a volume in a given number of parts along a direction, starting from an offset.

- `CHNAME` (`CHARACTER*4`) a unique name for the volume to be generated by subdivision of the mother volume;
- `CHMOTH` (`CHARACTER*4`) volume that has to be subdivided;
- `NDIV` (`INTEGER`) number of divisions into which the mother volume is to be divided;
- `IAXIS` (`INTEGER`) *axis* of the division.
- `CO` (`REAL`) offset where division should start – this value can be in centimeters or degrees according to the value of `IAXIS`;
- `NUMED` (`INTEGER`) medium number of the divisions – this can be different from the one of the mother, as the division cells may leave a portion of the mother undivided (see below) – if `NUMED` ≤ 0 the medium of the mother;

The divisions start at the user specified coordinate value and extend to the end of the volume. The range from this offset to the upper coordinate limit of the mother volume will be divided into the supplied number of cells. In the case of ϕ 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. Specifying an origin for the division, in this case, just moves the division boundaries. This can be useful to avoid a rotation. In all other cases the search routines will assume that a point is in the mother if the coordinate value is less than the value of the supplied offset.

```
CALL GSDVT2 (CHNAME,CHMOTH,STEP,IAXIS,CO,NUMED,NDVMX)
```

Divide a volume along a direction with a given step starting from an offset.

- `CHNAME` (`CHARACTER*4`) a unique name for the volume to be generated by subdivision of the mother volume;
- `CHMOTH` (`CHARACTER*4`) volume that has to be subdivided;
- `STEP` (`REAL`) size of the divisions – this value can be in centimeters or degrees according to the value of `IAXIS`;
- `IAXIS` (`INTEGER`) *axis* of the division;
- `CO` (`REAL`) offset where division should start – this value can be in centimeters or degrees according to the value of `IAXIS`;
- `NUMED` (`INTEGER`) medium number of the divisions – this can be different from the one of the mother, as the division cells may leave a portion of the mother undivided (see below) – if `NUMED` ≤ 0 the medium of the mother;
- `NDVMX` (`INTEGER`) expected (maximum) number of divisions – if ≤ 0 or > 255, 255 is assumed.

The division start at the user-specified coordinate value and extend to the end of the volume. The range from origin to upper coordinate limit of the mother volume is divided in sections of the user supplied step. If the step is such that the range of the mother cannot be filled with cells, the largest possible number of cells is created. The excess space up to the end of the mother volume will be assumed to belong to the mother.

In the case of ϕ division of a complete tube or cone, the whole 360 degrees will be filled with slices, no matter what the origin is. Specifying an origin for the division, in this case, just moves the division boundaries. This can be useful to avoid a rotation.

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.

Origin : R.Brun,F.Bruyant,A.McPherson
Revision :
Documentation :

Submitted: 01.11.83
Revised: 18.11.93

The volume data structure – JVOLUM

1 The data structure JVOLUM and JGPAR

The meaning of the variables in the data structure JVOLUM shown in fig. 29 is the following:

ISEARC search flag [GEOM400], [GEOM410]:
- $=$0 volume positioning order or ordering by GSNEXT/GSNEAR;
- $<$0 binary search as defined by GSORD;
- $>$0 user ordering GSUNEA/GUNEAR;

ISHAPE system shape number (see [GEOM050];
NIN number of volumes contained in the mother volume – if negative the volume is divided;
NMED medium number for the volume;
NPAR number of shape parameters;
NATT number of attributes;
PAR array of shape parameters;
IAT array of attributes;
IAXIS direction of the division (see [GEOM130]);
IVO system volume number;
NDIV number of divisions (-NDVMX, if computed dynamically, see [GEOM140]);
C0 coordinate value at which the division starts;
STEP coordinate step for the division;
NR user copy number;
IROT rotation matrix number defining the orientation of the volume with respect to the mother reference system;
X,Y,Z position of the volume with respect to the mother reference system;
KONLY ONLY/MANY flag, see [GEOM110] for more information;

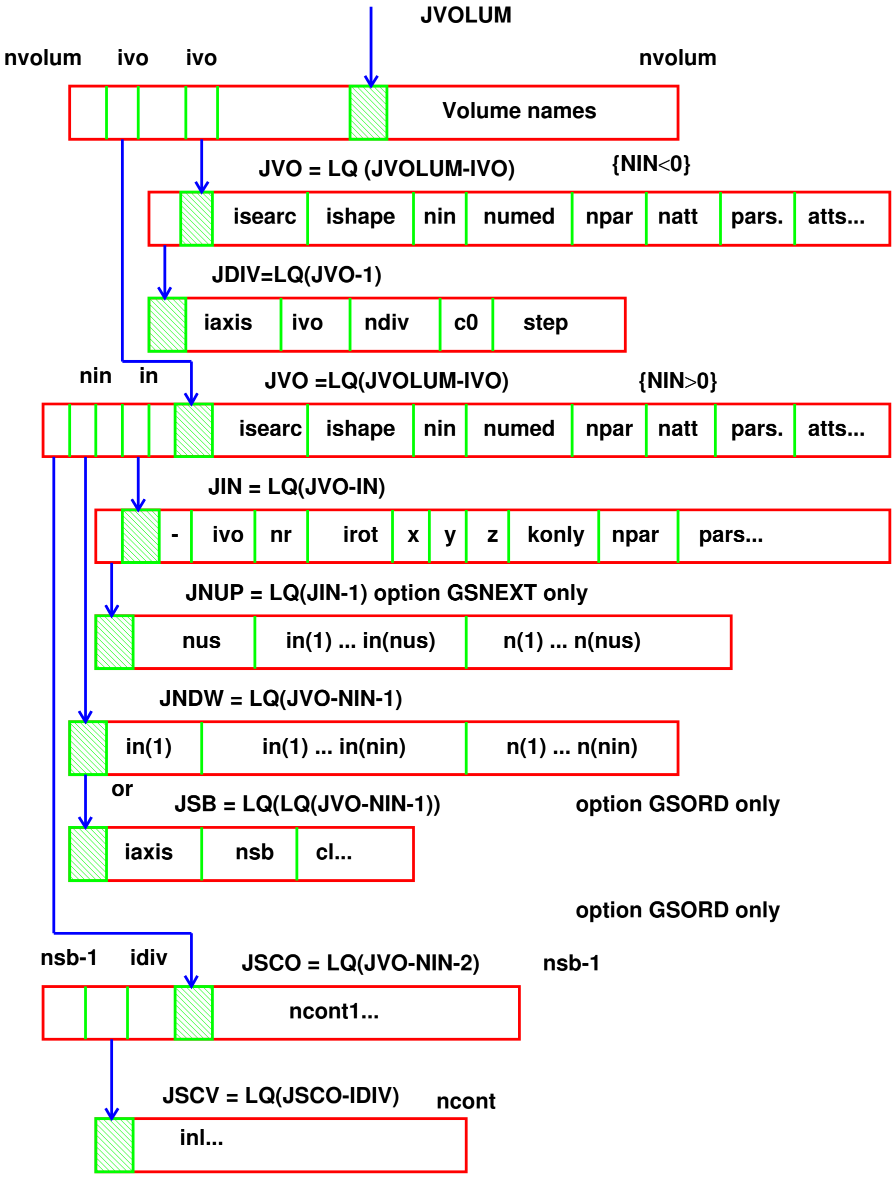


Figure 29: Example of geometrical tree structure

Origin : R.Brun, F.Carena **Submitted:** 01.06.83
Revision : **Revised:** 14.12.93
Documentation :

Rotation matrices

The relative position of a volume inside its mother is expressed in GEANT by a translation vector and a rotation matrix which are arguments of the routines GSPOS and GSPOSP. The rotation matrix expresses the transformation from the Mother Reference System to the Daughter Reference System.

A rotation matrix is described to GEANT by giving the polar and azimuthal angles of the axes of the DRS (x', y', z') in the MRS via the routine GSROTM.

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

IROT (INTEGER) number of the rotation matrix;
THETA1 (REAL) polar angle for axis x';
PHI1 (REAL) azimuthal angle for axis x';
THETA2 (REAL) polar angle for axis y';
THI2 (REAL) azimuthal angle for axis y';
THETA3 (REAL) polar angle for axis z';
PHI3 (REAL) azimuthal angle for axis z'.

Stores rotation matrix IROT in the data structure JROTM. If the matrix is not orthonormal, it will be corrected by setting $y' \perp x'$ and then $z' = x' \times y'$. A warning message is printed in this case.

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

Examples of use

The unit matrix is defined in the following way:

$$\left.\begin{array}{ccc} x' & \| & x \\ y' & \| & y \\ z' & \| & z \end{array}\right\} \Rightarrow \left\{\begin{array}{ccccccc} \theta_1 & = & 90^\circ & ; & \phi_1 & = & 0^\circ \\ \theta_2 & = & 90^\circ & ; & \phi_2 & = & 90^\circ \\ \theta_3 & = & 0^\circ & ; & \phi_3 & = & 0^\circ \end{array}\right.$$

This is just an example. There is in fact no need to define a unit rotation matrix. Giving the value 0 to the rotation matrix number in the call to GSPOS and GSPOSP is equivalent to a positioning without rotation and it improves tracking performance.

The result of a 90° counterclockwise rotation around z, followed by a 90° counterclockwise rotation around the new x is a cyclic shift of the axes: $x \rightarrow z'$, $y \rightarrow x'$, $z \rightarrow y'$. This is expressed by the following rotation matrix:

$$\left.\begin{array}{ccc} x' & \| & y \\ y' & \| & z \\ z' & \| & x \end{array}\right\} \Rightarrow \left\{\begin{array}{ccccccc} \theta_1 & = & 90^\circ & ; & \phi_1 & = & 90^\circ \\ \theta_2 & = & 0^\circ & ; & \phi_2 & = & 0^\circ \\ \theta_3 & = & 90^\circ & ; & \phi_3 & = & 0^\circ \end{array}\right.$$

Sometimes the rotation matrix is known or it can be constructed. In this case the arguments to the routine GSROTM can be calculated with the help of the routine GFANG in the following way:

```
      DIMENSION ROTMAT(3,3), ROWMAT(3), PHI(3), THETA(3)
      LOGICAL ROTATE
      .
      .
      .
      DO 10 I=1,3
         ROWMAT(1) = ROTMAT(I,1)
         ROWMAT(2) = ROTMAT(I,2)
         ROWMAT(3) = ROTMAT(I,3)
         CALL GFANG(ROWMAT,COSTH,SINTH,COSPH,SINPH,ROTATE)
         THETA(I) = ATAN2(SINTH,COSTH)
         PHI(I)   = ATAN2(SINPH,COSPH)
   10 CONTINUE
      .
      . {\sl Transform to degrees}
      .
      CALL GSROTM(IROT,THETA(1),PHI(1),THETA(2),PHI(2),THETA(3),PHI(3))
```

CALL **GPROTM** (IROT)

Prints the rotation matrix elements and angles.

IROT (INTEGER) rotation matrix number: if IROT=0 all rotation matrixes will be printed, if IROT<0, matrix number |IROT| will be printed without header information.

Origin : R.Brun
Revision :
Documentation :

Submitted: 01.11.78
Revised: 14.12.93

The rotation matrix data structure JROTM

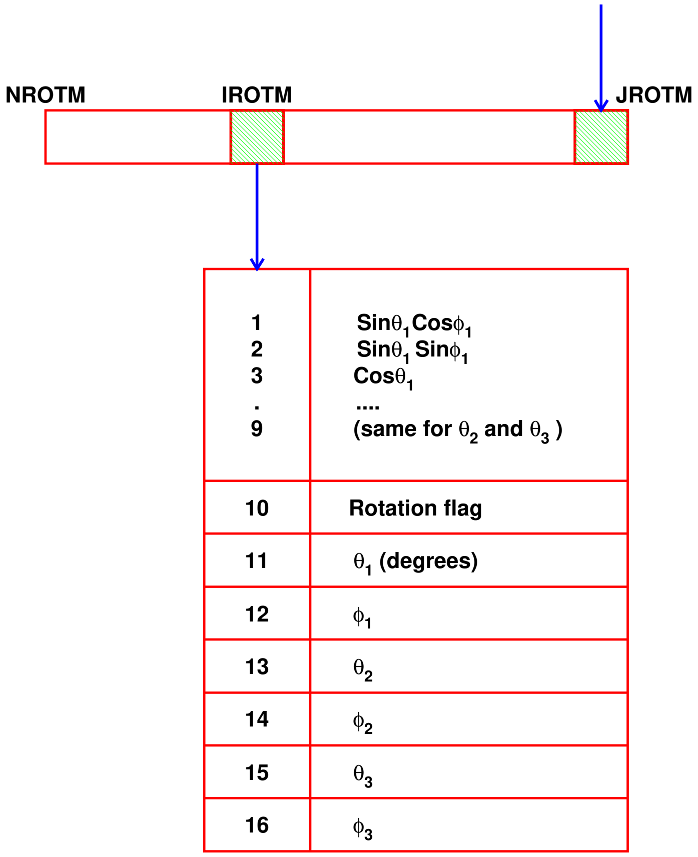


Figure 30: Layout of the JROTM data structure

`JR = LQ(JROTM-IROTM)` is the pointer to the bank of the rotation matrix `IROTM`;

The rotation flag is computed by `GSROTM` to recognise simple rotation configurations. In particular it is set to 0 for the unit matrix.

Origin : R.Brun, A.C.McPherson, F.Bruyant
Revision : S.Giani
Documentation :

Submitted: 18.12.83
Revised: 14.12.93

Finding in which volume a point is

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

X (REAL) array of dimension 3 with the coordinates in the MRS;
NUMED (INTEGER) medium number, if this is zero the point is outside the detector.

Searches the geometrical tree structure to find in which volume the point X is. The tracking medium is returned in NUMED, and the common /GCVOLU/ is updated.

GMEDIA uses the geometry data structure to conduct its search starting its search from the last volume where a point was found. If no previous search has been conducted, the first volume is used as a starting point.

If the point is not inside the current volume, GMEDIA looks in its mother and so on, until it finds a volume which contains the point. It then looks at the contents of this object and so on until the point is in a volume but in none of its contents (if any).

If this *downward* search terminates in a 'MANY' object, GMEDIA looks for another candidate. See [GEOM110] for a description of the 'MANY' volumes tratment.

The *physical* geometrical tree from the first volume to the current one is stored in the common block GCVOLU (see [ZZZZ010]) and in the structure JGPAR (see [GEOM199]).

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

X (REAL) array of dimension 3 with the coordinates in the MRS;
NUMED (INTEGER) medium number, if this is zero the point is outside the detector.

This routine performs the same function than GMEDIA, but it uses the dynamical information of the particle history to speed-up the search:

- when INWVOL=2 (common /GCTRAK/) the particle just came out of a volume. In this case, if INFROM (common /GCVOLU/) is positive, it is interpreted by GTMEDI as the number IN of the content just left, inside the mother volume where the point X is assumed to be. This content will not be searched again.
- the daughter of the current volume which limits the geometrical step of the particle (i.e. where the particle would be heading moving along a straight line) is searched first. INGOTO (common /GCVOLU/) is set by GTNEXT, to transmit the information on this one volume which has limited the geometrical step SNEXT as follows:

  >0 INth content;
  $=0$ current volume;
  <0 -NLONLY, with NLONLY defined as the lowest 'ONLY' level up in the tree which is an ancestor of the 'MANY' volume where the point X is.

```
CALL GINVOL (X,ISAME*)
```

X (REAL) array of dimension 3 with the coordinates in the MRS;
ISAME (INTEGER) return flag.

Checks if particle at point X has left current volume. If so, returns ISAME = 0 and prepares information useful to identify the new volume entered, otherwise, returns ISAME = 1.

Origin : R.Brun, A.C.McPherson
Revision : S.Giani
Documentation :

Submitted: 01.06.83
Revised: 15.12.93

Finding distance to next boundary

```
CALL GNEXT (X,SNEXT*,SAFETY*)
```

```
CALL GTNEXT (X,SNEXT*,SAFETY*)
```

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

X (REAL) array of 6 of coordinates and direction cosines;
SNEXT (REAL) distance to the nearest volume boundary along the particle direction;
SAFETY (REAL) *safety* distance, that is smaller distance to any boundary;

This routine evaluates the two *distances* which are needed by the GEANT tracking routines. GTNEXT and GNEXT perform the same function, but GNEXT is a static routine which can be called by the user, while GTNEXT is the routines used internally by GEANT during tracking, and it should not be called by the user.

If INFROM (common /GCVOLU/) is different from 0, GTNEXT interprets it as the daughter out of which the particle just came, and uses the list of daughters stored with that volume, possibly modified by GSNEXT/GSNEAR to calculate the distance to the next boundary.

The first action of GTNEXT is to calculate the SAFETY distance. If this is larger than the current step candidate, no other calculation is performed and the IGNEXT flag (common /GCTRAK/) is set to 0, indicating that no change of volume is occurring at the end of the current step. If the step is smaller than safety, then SNEXT is computed. If the step is smaller than SNEXT, again there will not be any change of volume during the step and IGNEXT is set to 0. If on the contrary the candidate step is larger than SNEXT, a change of volume will occur at the end of the step, and IGNEXT is set to 1 and INGOTO (common /GCTRAK/) is set to the number of the daughter where the particle is entering, if any.

Charged particles in magnetic field are transported with a similar logic. However, even if the candidate step is smaller than SNEXT, the particle can still cross into another volume due to its bent path. When tracking in magnetic field, after every step greater than SAFETY it is checked whether the particle is still in the same volume. If this is not the case, the step is divided by two and transport is tried again. Conversely a charged particle in magnetic field may still be in the current volume even after having travelled the distance to the nearest boundary along a *straight* line. So boundary crossing is declared only when IGNEXT is different from 0 and the difference between the real trajectory and the bent one is smaller than EPSIL (common /GCTMED/).

Origin : R. Brun
Revision :
Documentation :

Submitted: 18.08.87
Revised: 15.12.93

Reference system transformations

```
CALL GMTOD (XM,XD*,IFLAG)
```

XM (REAL) array of 3 containing the input position (IFLAG=1) or direction cosines (IFLAG=2) in the Master Reference System;

XD (REAL) array of 3 containing the output position (IFLAG=1) or direction cosines (IFLAG=2) in the (current) Daughter Reference System;

IFLAG (INTEGER) transformation flag:

- 1 input is a position, it must be translated and rotated;
- 2 input is a direction, it must be rotated.

This routine transform coordinates or directions from the MRS to the reference system of the current volume. The /GCVOLU/ must be properly filled, either by the tracking routines, by GMEDIA or by GLVOLU.

```
CALL GDTOM (XD,XM*,IFLAG)
```

Performs the inverse operation to GMTOD.

Origin : R.Brun,F.Bruyant,A.C.McPherson **Submitted:** 16.12.83
Revision : S.Egli **Revised:** 15.12.93
Documentation :

Pseudo-division of a volume

```
CALL GSORD (CHNAME,ICORD)
```

CHNAME (CHARACTER*4) name of the volume;

ICORD (INTEGER) direction of the pseudo-divisions:

- 1 x;
- 2 y;
- 3 z;
- 4 cylindrical R ($\sqrt{x^2+y^2}$);
- 5 spherical ρ ($\sqrt{x^2+y^2+z^2}$);
- 6 ϕ, azimuthal angle;
- 7 θ, polar angle with respect to the z axis.

This routine sets the search flag (Q(JVO+1)) of volume CHNAME to -ICORD. When the definition of the geometry is complete and GGCLOS is called, this flag is interpreted as a request to order the content of CHNAME along *axis* ICORD. This operation is performed by the routine GGORD. GGORD computes the limits of each of the contents along the given coordinate axis (see [GEOM001]), and prepares the lists of contents in each of the sections defined by the neighbouring coordinate. The JVOLUM structure is extended, for the mother volume, with banks which contains the list of boundaries and the lists of contents, so as to permit a binary search to access the contents of interest. The coordinates are in the local system of the mother volume. The routine GGORD will not be called if the number of contents exceeds 500.

The actual effect of this routine depends on the setting of the IOPTIM variable in the common /GCOPTI/. IOPTIM is controlled by the data record OPTI or the interactive command with the same name. The meaning of the different values of IOPTIM is the following:

- <0 no call to GGORD will be made, irrespective of the value of ISEARCH;
- 0 GGORD will be called for those volumes for which GSORD has been called;
- 1 for all volumes with contents for which neither GUSEAR nor GSORD has been called, the routine GGORDQ will be called;
- 2 GGORDQ is called for all volumes with contents for which GUSEAR has not been called.

GGORDQ orders the contents along all the possible axes and choses the ordering which provides the lowest number of volumes per division.

Origin : F.Bruyant, A.C.McPherson **Submitted:** 16.12.83
Revision : **Revised:** 14.12.93
Documentation : F.Carminati, M.Lefebvre

Ordering the contents of a volume

In the case of a mother volume containing a large number of daughters, tracking can be rather slow. This is due to the fact that each time GEANT requires to know in which daughter it is or will be tracking, it search through the whole list of daughter volumes. This is done by GEANT creating for every daughter volume a list which contain pointers to all the daughters of the same mother.

Clearly this may be avoided, because at any step of tracking, the coordinates and direction cosines of the current step are known. From this and the knowledge of the geometry, a restricted list of daughter volumes to be searched can be built. This can be accomplished in GEANT in two ways, which are described in this chapter.

Note: the user must be aware that the following routines alter the default search list of daughters of a given mother. A user mistake can cause wrong transport because GEANT does not make any check on the correctness of the list of neighbours provided by the user.

Static ordering

```
CALL GSNEXT (CHMOTH,IN,NLIST,LIST)
```

CHMOTH (CHARACTER*4) name of the mother volume to be ordered;
IN (INTEGER) number of the content for which a list is established;
NLIST (INTEGER) number of neighbours to be considered during tracking;
LIST (INTEGER) list of neighbours to volume IN.

This routine stores a given ordered LIST of NLIST daughter volumes to search when leaving the INth daughter of the mother volume CHMOTH.

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

```
CALL GSNEAR (CHMOTH,IN,NLIST,LIST)
```

CHMOTH (CHARACTER*4) name of the mother volume to be ordered;
IN (INTEGER) number of the content for which a list is established;
NLIST (INTEGER) number of neighbours to be considered during tracking;
LIST (INTEGER) list of neighbours to volume IN.

This routine stores a given ordered LIST of NLIST daughter volumes to search when leaving the INth daughter of the mother volume CHMOTH.

If LIST(1)=0 the particle is back into the mother when leaving the INth daughter. This means that the INth is not contiguous to any other daughter or to the boundary of the mother.

If IN=-1, the mother does not have contents contiguous to its boundaries (status bit 4 set in mother volume bank for action in GGCLOS).

If IN=0 for each content GSNEAR sets LIST(1)=0.

GSNEAR must be called after all contents have been position (except when IN=-1)

Dynamic ordering

The list of neighbours to search when exiting from the IN^{th} content may depend also on the direction and position of the particle. In case where it is necessary, for performance reasons, to exploit also this information, GEANT offers the possibility to the user to build a dynamic search list.

```
CALL GSUNEA (CHNAME,ISEARC)
```

CHNAME (CHARACTER*4) name of the volume where the user search has to be activated;

ISEARCH (INTEGER) specifies the kind of search list to be used: a positive value must be specified with this routine to activate user search lists.

This routine should be called once for every volume where user volume search is activated.

```
CALL GUNEAR (ISEARC,ICALL,XC,JNEAR)
```

ISEARCH (INTEGER) number associated to the volume in which the user search is used, it is the same number set by the user with GSUNEA;

ICALL (INTEGER) type of question that the list of volumes must answer:

- 1 GMEDIA-like call, where am I?
- 2 GTNEXT-like call, where can I go?

XC (REAL) array of 6 containing the position and the direction cosines of the particle (x, y, z, p_x/p, p_y/p, p_x/p);

JNEAR (INTEGER) pointer to the volume list bank which has to be filled by the user;

The list of volumes where GEANT has to search to answer the question specified by ICALL is returned by the user starting at Q(JNEAR+1. GEANT will only look at the volumes specified by the user and in the order in which they appear in the list. Daughters are numbered from 1 to N according to the order with which they have been positioned in the mother. The list should be filled in the following way:

IQ(JNEAR+1) = N, number of volumes in the list
IQ(JNEAR+1+1) = number of the first daughter to search
IQ(JNEAR+1+2) = number of the second daughter to search
.
.
.
IQ(JNEAR+1+N) = number of the N^{th} daughter to search

The user should be aware that this routine is called very often, almost at every tracking step, so it should be coded with the maximum efficiency in mind. An example of GUNEAR could be the following:

```
      SUBROUTINE GUNEAR(ISEARC,ICALL,XC,JNEAR)
*---             Make sure to add GEANT main store
+SEQ, GCBANK.
      DIMENSION XC(6), MYLIST(100)
*---             Executable code
      IF(ISEARC.EQ.1) THEN
*---             Build a list using XC and ISEARC for a GMEDIA type call
*---             Put all the daughters where the particle may be in
         MYLIST(1) = ....
         .
         .
         .
         MYLIST(N) = ....
      ELSE
```

```
*---            Build a list using XC and ISEARC for a GTNEXT type call
*---            Put all the daughters where the particle may be going
         MYLIST(1) = ....
         .
         .
         .
         MYLIST(N) = ....
      ENDIF
*---            Return the information to GEANT
      DO 10 I=1,N
         IQ(JNEAR+1+I) = MYLIST(I)
  10  CONTINUE
      IQ(JNEAR+1) = N
*---            End of GUNEAR
      END
```

Origin : R.Brun, P.Zanarini **Submitted:** 15.08.83
Revision : **Revised:** 15.12.93
Documentation :

Volume attributes

`CALL GSATT (CHNAME,CHIATT,IVAL)`

CHNAME (CHARACTER*4) volume name;
CHIATT (CHARACTER*4) attribute to be set;
IVAL (INTEGER) value to which the attribute is to be set.

Changes the attribute CHIATT of the volume called CHNAME to the value IVAL. The names and meaning of the attributes and their allowed values are:

Number	Name	Description
1	WORK	0 inactive volume 1 active volume
2	SEEN	-2 only the volume is visible, but none of its descendants -1 the volume is not visible together with all its descendants 0 the volume is not visible 1 the volume is visible
3	LSTY	line style parameter (see [XINT002])
4	LWID	line width parameter (see [DRAW400])
5	COLO	area filling colour (see [XINT002])
6	FILL	area filling resolution (see [DRAW400])
7	SET	set number associated to the volume
8	DET	detector number associated to the volume
9	DTYP	detector type associated to the volume (1,2)
10	NODE	dummy

`CALL GFATT (CHNAME,CHIATT,IVAL*)`

Returns in IVAL the attribute CHIATT of the volume CHNAME. The arguments have the same meaning than for the routine GSATT.

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

CHNAME (CHARACTER*4) volume name;
NR (INTEGER) copy number;
INTEXT (INTEGER) type of volume parameters requested:
  0 user parameters;
  1 internal parameters;
NPAR (INTEGER) number of parameters returned;

`NATT`	(`INTEGER`) number of attributes returned;
`PAR`	(`REAL`) array of parameters;
`ATT`	(`REAL`) array of attributes.

Returns parameters `PAR(1...NPAR)` and attributes `ATT(1...NATT)` for the volume `CHNAME` with copy number `NR`.

Origin : F.Bruyant, A.McPherson
Revision :
Documentation :

Submitted: 16.12.83
Revised: 15.12.93

User initialisation of the common block /GCVOLU/

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

NLEV (INTEGER) number of levels to fill;
LNAM (INTEGER) array of NLEV volume names stored via the UCTOH routine;
LNUM (INTEGER) array of NLEV volume copy (or division) numbers;
IER (INTEGER) returns code different from 0 in case of error, a diagnostic is also printed;

This routine fills the volume parameters in common /GCVOLU/ for a physical tree, specified by the lists LNAM and LNUM of volumes names and numbers, and for all its ascendants up to level 1.

The routine is optimised and does not re-compute the part of history already available in GCVOLU. This means that if it is used in user programs outside the usual framework of the tracking, the user has to initialise to zero NLEVEL in common GCVOLU.

An example of use of GLVOLU in this context could be to find the position and direction of a particle in the local coordinate system of a volume:

```
      .
      .
+SEQ,GCVOLU
      DIMENSION LNAM(15), LNUM(15), POS(3), DIR(3), POSL(3), DIRL(3)
      .
      .
      CALL UCTOH('MOTH',LNAM(1),4,4)
      LNUM(1) = 1
      CALL UCTOH('CAL1',LNAM(2),4,4)
      LNUM(2) = 2
      CALL UCTOH('MOD1',LNAM(3),4,4)
      LNUM(3) = 5
      CALL UCTOH('CHAM',LNAM(4),4,4)
      LNUM(4) = 18
*---
      NLEVEL  = 0
      CALL GLVOLU(4,LNAM,LNUM,IER)
*---
      CALL GMTOD(POS,POSL,1)
      CALL GMTOD(DIR,DIRL,2)
      .
      .
```

Origin : R.Brun **Submitted:** 16.12.83
Revision : **Revised:** 16.12.93
Documentation :

Medium search statistics

In order to understand the behaviour and the performance of the geometry defined by the user, it is sometimes very useful to accumulate the statistic of the various geometrical routines called. This is done automatically by `GEANT` if the statistical package is activated via the data record `STAT 1`. In this case the routine `GBSTAT` is called by `GGCLOS`, `GFSTAT` is called by the tracking routines `GINVOL`, `GTMEDI`, `GTNEXT`, `GTRACK` and `GMEDIA` to accumulate the statistics and `GPSTAT` is called by `GLAST`.

`CALL` **GBSTAT**

Creates the data structure `JGSTAT` in order to accumulate statistics for the number of calls to routines `GINVOL`, `GTMEDI`, `GTNEXT`, `GTRACK` and `GMEDIA` and the total number of steps per volume.

`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.

`CALL` **GFSTAT**

Fills banks for volume statistics.

An example of output is the following:

```
**************************************************************************************************************
*                                                                                                            *
*                                         V O L U M E   S T A T I S T I C S                                  *
*                                                                                                            *
**************************************************************************************************************
*  VOLUME  *        GINVOL        *        GMEDIA        *        GTNEXT        *        GTMEDI        * NSTEPS  *
*   NAME   *  TOTAL   *  NLEVEL   *  TOTAL   *  NLEVEL   *  TOTAL   *  NLEVEL   *  TOTAL   *  NLEVEL   * NLEVEL  *
**************************************************************************************************************
*   CAL0   *        0 *         0 *    43590 *         4 *   294743 *       250 *   246447 *       971 *      250 *
*   CAL1   *        0 *         0 *    43056 *         0 *   284132 *         0 *   236482 *         0 *        0 *
*   CAL2   *        0 *         0 *      524 *         0 *     9617 *         0 *     8283 *         0 *        0 *
*   CAL3   *        0 *         0 *        0 *         0 *        9 *         0 *        8 *         0 *        0 *
*   HOD1   *        0 *         0 *    43056 *        21 *   284132 *     42829 *   236482 *     42807 *    42848 *
*   HOD2   *        0 *         0 *      524 *         0 *     9617 *      1570 *     8283 *      1570 *     1570 *
*   HOD3   *        0 *         0 *        0 *         0 *        9 *         2 *        8 *         2 *        2 *
*   SHIL   *        0 *         0 *     4935 *      4935 *    54736 *     54736 *    44480 *     44480 *    58247 *
*   URPL   *        0 *         0 *    35932 *     35932 *    52803 *     52803 *    22571 *     22571 *    89953 *
*   CHA1   *        0 *         0 *     1703 *         0 *    90507 *         0 *    86717 *         0 *        0 *
*   TUB1   *        0 *         0 *     1703 *      1670 *    90507 *     62174 *    86717 *     58423 *    62700 *
*   GAS1   *        0 *         0 *       33 *        33 *    28333 *     28333 *    28294 *     28294 *    28350 *
*   EPO1   *        0 *         0 *      920 *       920 *    47007 *     47007 *    43032 *     43032 *    47742 *
*   COPL   *        0 *         0 *        0 *         0 *        2 *         2 *        2 *         2 *        2 *
*   CHA2   *        0 *         0 *       75 *         0 *     5037 *         0 *     4295 *         0 *        0 *
*   TUB2   *        0 *         0 *       75 *        74 *     5037 *      4079 *     4295 *      3338 *     4159 *
*   GAS2   *        0 *         0 *        1 *         1 *      958 *       958 *      957 *       957 *      958 *
**************************************************************************************************************
```

Origin : A.C.McPherson **Submitted:** 16.12.83
Revision : F.Bruyant, R.Brun **Revised:** 16.12.93
Documentation :

End of geometry definition

CALL **GGCLOS**

This routine should be called after all volumes, positions, hits and digits have been defined. Its action is confined to clean up the volume banks and to take any action required by the ordering techniques.

GGCLOS performs all the requested geometry optimisation, included the creation of pseudo-divisions in volumes flagged by GSORD. It performs the local development of tha last leaves of the geometrical tree, where applicable.

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) [HITS001].

GGCLOS must be called also after reading a geometry data structure from disk.

Origin : J.Vuoskoski, N. Høimyr **Submitted:** 22.03.94
Revision : **Revised:** 23.09.94
Documentation :

The CADINT Interface

1 Introduction

The engineering work in HEP for design, analysis and manufacturing of detectors requires Computer Aided Engineering (CAE) tools. To ensure coherent design the same detector descriptions should be used by physicists, engineers and, eventually, manufacturers. Efficient exchange of data between CAE tools and `GEANT` becomes crucial.
The CADINT interface allows to export detector models from `GEANT` to Computer Aided Design (CAD) systems. CADINT outputs detector models in the SET (Standard d'Echange et de Transfert) [1] neutral file format. A detector model is written as an assembly of solid volumes, with respect to a global coordinate system. The geometric representation used is Constructed Solid Geometry (CSG) [2].
A `GEANT` detector geometry exported through CADINT into a CAD system can be used to check the coherence of the model used in simulation. The CADDFAS service at CERN can be used to convert SET files into IGES (Initial Graphics Exchange Specification, an American standard for exchange of CAD data) [3] format, which is readable by most CAD systems. For more information about the CADDFAS service, send an e-mail to *caddfas@cadd.cern.ch* with subject: *info*

2 SET File Format

SET is a French standard to exchange and archive CAD data. It was developed as a neutral file format for exchanging data between different CAD systems at Aerospatiale in 1983. The aim was to develop a more reliable alternative to IGES. It became an official French standard, Afnor Z68-300, SET, in 1985. SET was revised and extended in 1989. The version (89-06) used in CADINT, supports wireframe, surface and solid entities. Entities for drafting and connectivity applications, as well as scientific data and FEM (Finite Element Method) modelling are also included. The philosophy behind the standard is that it should be able to handle the complete information that may occur. It is also considered to be important to have an unambiguously defined format, that is compact in size, and is flexible enough to handle future demands from the CAD/CAM industry.
The structure of SET is based upon a three level hierarchical structure, which consists of data assemblies, data blocks, and data sub-blocks. Information that is common to several blocks or assemblies, is stored in a so-called dictionary.

- **Assembly:** A SET data assembly is a collection of data defining a certain piece of information, such as a mechanical part. A SET file can contain one or more data assemblies;
- **Block:** In the nomenclature of the SET files, the blocks are identified by an @, followed by the number of the block identifying its type. This number is followed by the block's reference number in the file. (i.e. in a file with *n* blocks, the block sequence numbers will be from 0 to *n*.) After this, sub-blocks and dictionary entries used in the block follows. A SET data block is an elementary entity, which consists of definition or control data that are used in different applications. Such entities could be geometric objects as points or lines, or other entities like matrices, drawings and views or SET file identifiers;
- **Sub-block:** A sub-block consists of an identifier, and a list of data that contributes to the description of the entity defined by the data block. The different parameters inside a sub-block, such as coordinates, are represented by their values. A sub-block has the identifier "#", followed by its type number, possible references to the dictionary, and parameters applicable in the sub-block;

- **Dictionary:** The dictionary is a set of predefined parameters in the specifications of the standard. They are accessible as dictionary entries which are assigned an identifier (a colon "**:**" and the dictionary number), and a precise meaning given by authorised and/or default values.

References among entities in a SET file are made by using pointers, either directly from the blocks, or via sub-blocks or dictionary entries. Pointers to other blocks are identified by an "**!**", followed by the sequence numbers of the blocks. SET has no unique mapping of sub-block types and dictionary entries into blocks. Several combinations are allowed for each different block. Definitions of possible combinations and guidelines for implementation are given in the SET standard [1].

3 Implementation

In the CADINT interface, a user can indicate a subtree of a detector by giving the name of a volume. All the contents of this volume will be written into a SET file. A user can also choose the contents by setting the daughter volumes visible or invisible using the visibility attribute. Only the visible volumes will be written into the SET file. In order to avoid repeated volumes, a number of divisions can be chosen. The rest of the divisions can be reproduced in a CAD system, if needed. The colors of volumes defined by the `GEANT` color attribute are also transmitted into SET files.

The CADINT interface writes the geometry of a `GEANT` detector model in the SET file format in two basic phases:

1. write the volume information;
2. write the position information.

The interface writes volumes into the SET format as CSG solids i.e. the volumes are created using extrusions, revolutions, ruled solids, rectangular parallelepipeds, etc. Divisions are written as normal volumes. Every division of a divided volume is a distinct volume in a SET file. An index is attached to the end of the name of each of the divisions in order to distinguish each division instance. The indexing of instances is reset in each division. An index is attached to the end of the volume name of multiple copies of normal volumes as well.

In SET format, a volume must be first written, after that it can be positioned. All volumes are positioned with respect to the global coordinate system. Thus, the tree structure in the SET file is flat. There will be overlaps between volumes, since no Boolean operations are used to define the tree structure. However, the tree structure is written into a separate file. The material information is written into this same text file.

The interface decodes the `JVOLUM` data structure and computes the necessary parameters if needed (negative parameters, dimensions of divisions, etc.). These computations are based on the `GEANT` drawing routines. The interface decodes a detector model starting from the global mother volume. After that it decodes the first daughter volume in the left side of the tree. The daughter volume can be a normal volume or a division. It continues decoding daughter volumes until it is at the bottom of the tree. After that the interface returns one level and decodes the next daughter of the current volume if any exists. In a case of a divided volume, the divisions are treated in the same way as the daughter volumes. The creation of the SET file is performed simultaneously.

3.1 Volume Information

The tree decoding routine transfers the volume information during decoding to the relevant shape routine which writes the volumes into the SET file. Every different shape has its own routine. For example, the simplest shape, `BOX`, is written in the SET format as follows:

```
@50,N1,:5,2#60,X,Y,Z
@302,N2#317,-DX,-DY,-DZ
@100,N3,:5,2,:9,'Name'#101,!N1,!N2
```

Where

- `@50` is the block number which defines a primitive solid;
- `N1,N2,N3` are the sequence numbers of each block;
- `:5,2` is the dictionary entry number which gives a total subordination to one block;
- `#60` is the sub-block number which defines the geometric parameters of a solid rectangular parallelepiped;
- `X,Y,Z` are the dimensions of the rectangular parallelepiped;
- `@302` is the block number for geometric transformation;
- `#317` is the sub-block number for translation;
- `DX,DY,DZ` are the coefficients of the translation;
- `@100` is the block number which defines a constructed solid;
- `:9` is the dictionary entry number for a name associated with the block;
- `Name` is the name of the volume;
- `#101` is the sub-block number for transformation operation.

The rectangular parallelepiped is first defined with the given parameters. Then the translation of coordinates is defined. The translation is necessary since `GEANT` and SET use origins in different places. Finally, the transformation operation is defined for a constructed solid which in this case is the translation of the coordinates of a primitive solid. The GEANT shapes are converted to SET as follows:

SHAPE	DEFINITION IN SET
BOX	rectangular parallelepiped
TRD1	ruled solid between 2 faces
TRD2	ruled solid between 2 faces
TRAP	ruled solid between 2 faces
TUBE	solid of revolution of a face
TUBS	solid of revolution of a face
CONE	solid of revolution of a face
CONS	solid of revolution of a face
SPHE	sphere
PARA	ruled solid between 2 faces
PGON	first segment ruled solid, other segments copied and rotated, finally boolean union for all the segments
PCON	solid of revolution of a face
ELTU	solid of linear extrusion of a face
HYPE	not implemented
GTRA	ruled solid between 2 faces
CTUB	solid of revolution and boolean subtraction by 2 half-spaces

3.2 Position Information

The position information of a volume is handled by a routine, which writes the coordinates and the rotation matrix. The tree decoding routine transforms the coordinates and the rotation matrix of a volume to the global coordinate system, and then transfers the data to the routine which writes the translation and rotation of a volume into the SET format as follows:

```
@302,N1#301,M1,M2,M3,M4,M5,M6,M7,M8,M9,XC,YC,ZC
@100,N2,:9,'Name'#101,!N1(of the shape),!N1
```

Where

- `@302` is the block number for geometric transformation;
- `N1,N2` are the sequence numbers of each block;
- `#301` is the sub-block number which defines the coefficients of the rotation-translation matrix;
- `M1...M9` are the coefficients of the rotation matrix;
- `XC,YC,ZC` are the components of the translation vector;
- `@100` is the block number which defines a constructed solid;
- `:9` is the dictionary entry number for a name associated with the block;
- `Name` is the name of the volume;
- `#101` is the sub-block number for transformation operation;
- `!N1` is the reference to sequence number of the mother volume.

The geometric transformation is first defined with the given parameters and then the transformation operation is defined for a constructed solid. The principle of writing divisions into the SET format is equivalent to the case of normal volumes.

3.3 Material and Tree Information

The material file is written into a separate file simultaneously with the SET file. It contains information on tracking medium, material and density for each defined volume. The `GEANT` tree is written into this same file. After the volume name there follows the number of daughters and daughter names. In a case of divided volume, the negative number of divisions is given after the name of the divided volume, and followed by the offset, step and the name of the division instance. An output of a material file of a `GEANT` example program:

```
 GEANT-SET MATERIAL LISTING FILE
 -------------------------------
 Materials in the geometry described in
 the .SET file:  gexam4.set

Volume name  Tracking media   Material         Density

CALO         1 AIR            1 AIR            0.12000001E-02
CAL1         1 AIR            1 AIR            0.12000001E-02
MOD1         1 AIR            1 AIR            0.12000001E-02
CAL2         1 AIR            1 AIR            0.12000001E-02
MOD2         1 AIR            1 AIR            0.12000001E-02
CAL3         1 AIR            1 AIR            0.12000001E-02
MOD3         1 AIR            1 AIR            0.12000001E-02
EPO1         4 CARBON         4 CARBON         0.22650001E+01
CHA1         6 BRASS          6 BRASS          0.85600004E+01
TUB1         6 BRASS          6 BRASS          0.85600004E+01
GAS1         5 GAS            5 ARG/ISOBU      0.21360000E-02

 GEANT TREE
 ----------
  The GEANT tree starting from the given volume

CALO    6  CAL1  CAL2  CAL3  EPO1  CHA1  EPO1
CAL1  -64  -0.48000000E+02   0.15000000E+01  MOD1
CAL2  -35  -0.43750000E+02   0.25000000E+01  MOD2
CAL3  -13  -0.24050001E+02   0.37000003E+01  MOD3
CHA1  -40  -0.25000000E+02   0.12500000E+01  TUB1
MOD1    6  SHIL  URPL  SHIL  EPO1  CHA1  EPO1
MOD2    4  SHIL  URPL  SHIL  CHA2
MOD3    2  COPL  CHA2
TUB1    1  GAS1
CHA2  -72  -0.23500000E+02   0.65277779E+00  TUB2
TUB2    1  GAS2

        ------ end of file -------
```

4 Usage

In order to carry out a successful transfer of a detector model, a user needs to be aware of certain limitations of the interface. The user is recommended to communicate with the engineers using CAD systems to establish a proper transfer methodology. If a CAD system is not capable of receiving the whole detector geometry in a single SET file, a user can transfer a detector model in smaller parts. These parts can then be joined together in the CAD system. Normally it is a good idea to just export one instance of each division in order to limit the number of exported volumes. The divisions can be reproduced in CAD systems. It is impossible to give an exact size-limit for a detector model which can be transferred in one file, because the limitation depends on the structure of the detector model.

The `GEANT` graphics package provides excellent capabilities for viewing parts of a detector model which is to be exported. Using the visibility attribute and the number of division instances, a user will be able to see what is to be exported. A user can use different colors in order to help to distinguish different volumes (eg division instances, etc.) in a CAD system. Dummy volumes, such as mothers made of air or vacuum should be suppressed using the visibility attribute, since the interface is currently not able to discard such information. A user must remember to specify the drawing parameters again if he wants to draw the detector model after using the interface. This is because the interface uses modified drawing routines.

A plot of the `GEANT` tree can be useful for engineers to understand the structure of the `GEANT` detector model. `GEANT` volume names are useful references regarding the material and tree information. The tree information is also provided in the material file, but a graphical presentation is often useful.

GEOM Bibliography

[1] French Standardization, Data exchange and transfer standard specification, Afnor 1989.

[2] Michael E. Mortenson, Geometric Modeling, John Wiley & Sons, Inc., 1985.

[3] US department of Commerce, Initial Graphics Exchange Specification, 1988.

GEOM Bibliography

Origin :
Revision :
Documentation : F.Bruyant

Submitted: 15.08.84
Revised: 17.12.93

The detector response package

1 Introduction

In the context of GEANT:

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

The detector response package consists of tools to store, retrieve or print the information relevant to hits and digitisation which is in the data structures JSET, JHITS and JDIGI. A few subroutines which may help the user to solve some of the usual digitisation problems 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 setups 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.

Two remarks can be made:

- during the life of an experiment, the stability of the format and content 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 further processing 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]), in particular with the concepts of logical and physical volume tree structure.

The user is required to classify into sets all sensitive detectors (defined as those volume defined as detector via GSDET/GSDETV) for which he wants to store hits in the data structure JHITS. 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 record SETS. The user can group together in one or in several sets detectors of the same or different types. For convenience, it is recommended to have at least one set for each main component of the setup, e.g. hadronic calorimeters, electromagnetic calorimeters, vertex chamber, etc.

A volume can be declared as a 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 in the physical setup, some additional information is needed for associating correctly the hits with the physical detectors.

When using GSDET the user has to enter the (shortest) list of volume **names** (the vector CHNMSV), which permits unambiguous identification of the path through the physical tree, even in the presence of multiple copies. This identification is obtained by specifying a list of volume **numbers** (the vector NUMBV), in a one to one correspondence with the list of volume names. This list, after packing, will constitute the identifier of the physical detector.

If GSDETV is used instead of GSDET then the routine GGDETV (called by GGCLOS) constructs the lists CHNMSV automatically and stores them in the structure JSET.

3 The user tools

The data structure JSET is built through calls to the routine GSDET or GSDETV which assign detectors to sets and define their parameters. After this, the following routines can be called, for each detector, to complete the structure:

GSDETH provides the parameters required for the storage of the hit elements in the data structure JHITS, such as the packing and scaling conventions;

GSDETD provides the parameters required for the storage of the digitisations in the structure JDIGI, such as the packing conventions;

GSDETU adds 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. 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 through 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;
- 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;
- 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 routine GFHITS (or GFDIGI) which extracts the information from the JHITS or JDIGI structures, and may be used to retrieve the geometrical characteristics of the given detector.

If the detectors have been defined by the routine GSDETV, the geometrical information can be retrieved by the routines GFPATH and GLVOLU. GFPATH prepares the lists CHNAM and LNUM required by GLVOLU ([GEOM001]), from the information preprocessed at initialisation time by the routine GGDETV and stored in the structure JSET

Origin : R.Brun,F.Bruyant
Revision :
Documentation :

Submitted: 01.11.83
Revised: 17.12.93

Sensitive DETector definition

`CALL GSDET (CHSET,CHDET,NV,CHNMSV,NBITSV,IDTYP,NWHI,NWDI,ISET*,IDET*)`

CHSET	(CHARACTER*4) set identifier, user defined;
CHDET	(CHARACTER*4) detector identifier, has to be the name of an existing volume;
NV	(INTEGER) number of volume descriptors;
CHNMSV	(CHARACTER*4) array of NV volume descriptors;
NBITSV	(INTEGER) array of NV, NBITSV(I) (I=1,...,NV) is the number of bits in which to pack the copy number of volume CHNMSV(I);
IDTYP	(INTEGER) detector type, user defined;
NWHI	(INTEGER) initial size of HITS banks;
NWDI	(INTEGER) initial size of DIGI banks;
ISET	(INTEGER) position of set in bank JSET;
IDET	(INTEGER) position of detector in bank JS=LQ(JSET-ISET).

Assigns detector CHDET to the set CHSET and defines its basic parameters.

Note: The vector CHNMSV (length NV) contains the list of volume names which permit unambiguous identification of all copies of volume CHDET [see example in [HITS110]. 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 [HITS110]. The detector type IDTYP is not used internally by GEANT and can be used to distinguish quickly between various kinds of detectors, in the routine GUSTEP for example.

`CALL GSDETV (CHSET,CHDET,IDTYP,NWHI,NWDI,ISET*,IDET*)`

The arguments of this routine are the same than the previous one, but NAMES, NBITSV will be computed by GGDETV called by GGCLOS) (see [HITS001]).

`CALL GFDET (CHSET,CHDET,NV*,CHNMSV*,NBITSV*,IDTYP*,NWHI*,NWDI*,ISET*,IDET*)`

Returns the parameters for detector CHDET of set CHSET, the arguments have the same meaning than for routine GSDET.

`CALL GPSETS (CHSET,CHDET)`

Prints SET and DETector parameters.

CHSET	(CHARACTER*4) set to be printed, if * prints all detectors of all sets;
CHDET	(CHARACTER*4) detector to be printed, if * prints all detectors of set CHSET.

Origin : F.Bruyant
Revision :
Documentation :

Submitted: 13.06. 85
Revised: 16.12.93

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 (CHSET,CHDET,CHALI,NWHI,NWDI,IALI*)
```

CHSET	(CHARACTER*4) set name;
CHDET	(CHARACTER*4) name of the detector for which an alias is being defined;
CHALI	(CHARACTER*4) alias name;
NWHI	(INTEGER) initial number of words of HITS banks;
NWDI	(INTEGER) initial number of words of DIGI banks;
IALI	(INTEGER) position of alias in bank JS=LQ(JSET-ISET).

Defines an alias CHALI for detector CHDET of set CHSET and enters it in the JSET structure as an additional detector in the corresponding set, at the position IALI. The new detector will be a copy at position IALI of the original detector CHDET, 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 CHALI. Several aliases can be defined for the same detector through calls to GSDETA.

Origin	: R.Brun		**Submitted:** 01.11.83
Revision	:		**Revised:** 17.12.93
Documentation	:		

DETector hit parameters

CALL **GSDETH** (CHSET,CHDET,NH,CHNAMH,NBITSH,ORIG,FACT)

CHSET	(CHARACTER*4) set name;
CHDET	(CHARACTER*4) detector name;
NH	(INTEGER) number of the components of a hit;
CHNAMH	(CHARACTER*4) array of NH names for the hit components;
NBITSH	(INTEGER) array of NH, NBITSH(I) (I=1,...NH) is the number of bits in which to pack the I^{th} component of the hit;
ORIG	(REAL) array of NH offset applied before packing the hits values;
FACT	(REAL) array of NH scale factors applied before packing the hits values;

Defines hit parameters for detector CHDET of set CHSET. The routine must be called at initialisation time once the geometrical volumes have been defined to describe the hit elements and the way to pack them in the data structure JHITS. The value of the hit before packing is transformed in the following way:

$$\texttt{VAL(I) = (HIT(I)+ORIG(I))} \times \texttt{FACT(I)}$$

Example

Assume an electromagnetic calorimeter ECAL divided into 40 PHI sections called EPHI. Each EPHI division is in turn divided along the z axis in 60 sections 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 quantities which should be stored for each hit are:

X	x position of the hit in the lead glass block ($-1000 < x < 1000$);
Y	y position of the hit in the lead glass block ($-1000 < y < 1000$);
Z	z position of the hit in the lead glass block ($-1000 < z < 1000$);
E	energy of the particle;
ELOS	the energy deposited;

In this scheme a hit could look like:

Element	Value
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 code to define the SET/DET/HIT information could be:

```
      CHARACTER*4 CHNMSV(3),CHNAMH(5)
      DIMENSION   NBITSV(3),NBITSH(5)
      DIMENSION   ORIG(5),FACT(5)
*---
      DATA CHNMSV/'EPHI','EZRI','BLOC'/
      DATA NBITSV/     6,     6,     3/
*---
      DATA CHNAMH /'X   ','Y   ','Z   ','E   ','ELOS'/
      DATA NBITSH /    16,    16,    16,    16,    16/
      DATA ORIG   / 1000., 1000., 1000.,    0.,    0./
      DATA FACT   /   10.,   10.,   10.,  100.,  100./
*---
      CALL GSDET ('ECAL','BLOC',3,CHNMSV,NBITSV,2,100,100,ISET,IDET)
      CALL GSDETH('ECAL','BLOC',5,CHNAMH,NBITSH,ORIG,FACT)
```

CALL **GFDETH** (CHSET,CHDET,NH*,CHNAMH*,NBITSH*,ORIG*,FACT*)

Returns the hit parameters for detector CHDET of set CHSET. All arguments are as explained above.

Origin : R.Brun
Revision :
Documentation :

Submitted: 01.11.83
Revised: 17.12.93

DETector Digitisation parameters

```
CALL GSDETD (CHSET,CHDET,ND,CHNMSD,NBITSD)
```

CHSET	(CHARACTER*4) set name;
CHDET	(CHARACTER*4) detector name;
ND	(INTEGER) number of elements per digitisation;
CHNMSD	(CHARACTER*4) array of ND names for the digitisation elements;
NBITSD	(INTEGER) array of ND, NBITSD(I) (I=1,...,ND) is the number of bits into which to pack the I^{th} element of the digit.

Defines digitisation parameters for detector CHDET of set CHSET. The routine must be called at initialisation, after the geometrical volumes have been defined, to describe the digitisation elements and the way to pack them in the data structure JDIGI.

Example

Following on from the example in [HITS110], we want to add the digitisation information to detector EPHI of set ECAL. The information which should be stored for each digitisation is the ADC pulse height in the lead glass block, so that a digitisation in this scheme could look like:

Element	Value
EPHI	12
EZRI	41
BLOC	3
ADC	789

The code to define the digitisation information could be:

```
CHARACTER*4 CHNMSD
DATA CHNMSD/'ADC '/
DATA NBITSD/16/

CALL GSDETD('ECAL','BLOC',1,CHNMSD,NBITSD)
```

```
CALL GFDETD (CHSET,CHDET,ND*,CHNMSD*,NBITSD*)
```

Returns the digitisation parameters for detector CHDET of set CHSET. All arguments as explained in GSDETD.

Origin : J.J.Dumont, W.Gebel **Submitted:** 01.11.83
Revision : **Revised:** 17.12.93
Documentation :

User detector parameters

```
CALL GSDETU (CHSET,CHDET,NUPAR,UPAR)
```

CHSET (CHARACTER*4) set name;
CHDET (CHARACTER*4) detector name;
NUPAR (INTEGER) number of parameters;
UPAR (REAL) array of NUPAR parameters.

Stores user parameters for detector CHDET of set CHSET. The routine must be called at initialisation after the geometry has been defined.

```
CALL GFDETU (CHSET,CHDET,NUPAR,NW*,UPAR*)
```

The parameters have the same meaning than in the previous routine, apart from:

NUPAR (INTEGER) size of the UPAR array;
NW (INTEGER) number of parameters returned in UPAR, NW $\leq$ NUPAR;

Returns the user parameters for detector CHDET of set CHSET.

Origin : R.Brun **Submitted:** 01.11.83
Revision : **Revised:** 16.12.93
Documentation :

The SET data structure JSET

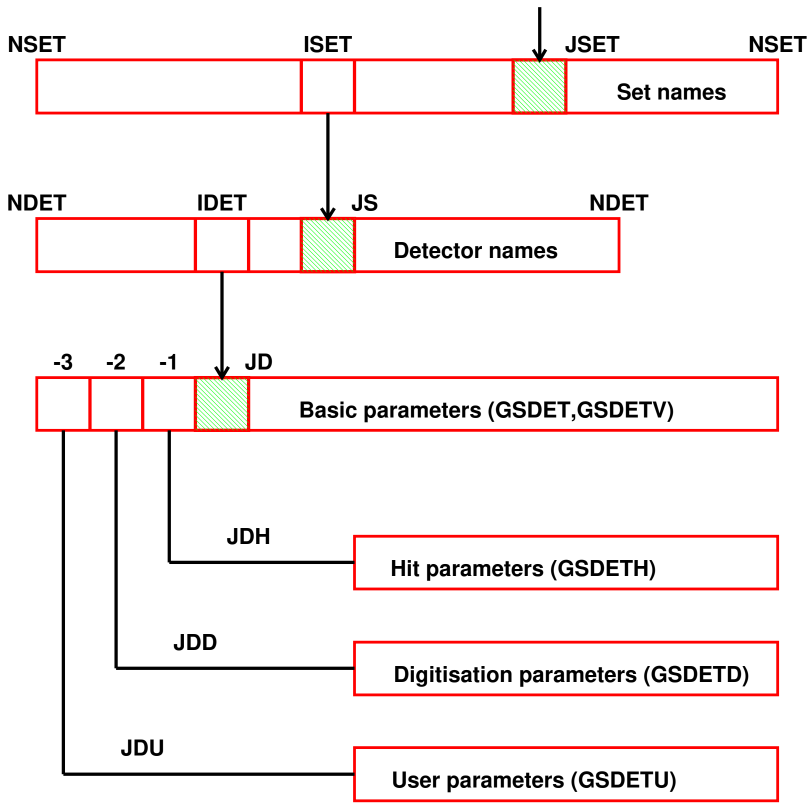


Figure 31: Example of geometrical tree structure

JS = LQ(JSET-ISET) pointer to detector set number ISET The JSET data structure is filled by GSDET, GSDETV, GSDETH, GSDETD, GSDETU and possibly by GSDETA.

Origin : R.Brun, F.Bruyant, W.Gebel, M.Maire
Revision :
Documentation :

Submitted: 01.11.83
Revised: 17.12.93

Routines to store and retrieve HITS

```
CALL GSAHIT (ISET,IDET,ITRA,NUMBV,HITS,IHIT*)
```

ISET	(INTEGER) set number (see below);
IDET	(INTEGER) detector number;
ITRA	(INTEGER) number of the track producing this hit
NUMBV	(INTEGER) array of volume numbers corresponding to list NAMESV of GSDET;
HITS	(REAL) array of values for current hit elements;
IHIT	(INTEGER) current hit number, if 0 the hit has not been stored.

Stores element values for current hit into the data structure JHITS. The values ISET, IDET and NUMBV can be found in the corresponding variables of common /GCSETS/. These values are set by the routine GFINDS every time that a particle is in a defined detector.

```
CALL GSCHIT (ISET,IDET,ITRA,NUMBV,HITS,NHSUM,IHIT*)
```

Same action as GSAHIT, but in case the detector identified by ISET, IDET and NUMBV contains already a hit for the same track, the routine will make a cummulative sum for the latest NHSUM elements of JHITS. The other previous elements of JHITS are replaced. That facility is particularly interesting in the case of hits generated into a calorimeter. No packing (i.e. 32 bits per hit element) should be requested for the last NHSUM hits in GSDETH and for these hits ORIG should be set to 0.

```
CALL GPHITS (CHSET,CHDET)
```

CHSET	(CHARACTER*4) set name, if '*' prints all JHITS banks of all sets;
CHDET	(CHARACTER*4) detector name, if '*' prints hits in all detectors of set CHSET.

Prints JHITS banks for detector CHDET of set CHSET.

```
CALL GFHITS (CHSET,CHDET,NVDIM,NHDIM,NHMAX,ITRS,NUMVS,
             ITRA*,NUMBV*,HITS*,NHITS*)
```

CHSET	(CHARACTER*4) set name;
CHDET	(CHARACTER*4) detector name;
NVDIM	(INTEGER) 1^{st} dimension of arrays NUMBV and NUMVS: 1≤NVDIM≤NV argument of GSDET;
NHDIM	(INTEGER) 1^{st} dimension of array HITS: 1≤NHDIM≤NH argument of GSDETH;
NHMAX	(INTEGER) maximum number of hits to be returned, this should be not larger than the second dimension of array NUMBV and HITS;
ITRS	(INTEGER) number of the selected track, if ITRS=0, all tracks are taken;
NUMVS	(INTEGER) is a 1-dimension array of NVDIM elements that contains the list of volume numbers which identify the selected detector, 0 is interpreted as 'all valid numbers';
ITRA	(INTEGER) is a 1-dim array of dimension NHMAX that contains on output for each hit the number of the track which has produced it;
NUMBV	(INTEGER) 2-dim array (NVDIM,NHMAX) that containis on output for each hit the list of volume numbers which identify the detector, all values set to 0 means that no more volumes are stored;
HITS	(REAL) 2-dim array (NHDIM,NHMAX) that containis NHITS hits;

NHITS	(INTEGER) number of hits returned, in case the total number of hits is greater than NHMAX, NHITS is set to NHMAX+1 and NHMAX hits are returned.

This rotine returns the hits produced by track ITRS (or by any track) in the detector CHDET identified by the list NUMVS belonging to set CHSET.

HITS(1,I)	is element 1 of hit number I;
NUMBV(1,I)	is volume number 1 of hit number I;
ITRA(I)	is the track number corresponding to hit number I;

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*)

ISET	(INTEGER) set number;
IDET	(INTEGER) detector number;
NUMBV	(INTEGER) array of numbers which identify uniquely detector number IDET;
NLEV	(INTEGER) number of elements filled of arrays LNAM and LNUM;
LNAM	(INTEGER) array of NLEV volume names stored in ASCII code in integers;
LNUM	(INTEGER) array of NLEV copy numbers.

This routine returns the list of volume names and numbers which identify the complete ancestry in the JVOLUM data structure of the volume corresponding to the detector number IDET in set number ISET and which is identified by the volume numbers NUMBV (see GFHITS).

GFPATH assumes that the detectors have been declared via GSDETV and not GSDET. The main use of GFPATH is to prepare the lists LNAM and LNUM required by the routine GLVOLU to fill the common /GCVOLU/. Once /GCVOLU/ is properly filled, it is possible to use the GEANT routines to transform from the local to the master reference system and so on.

Origin : R.Brun
Revision :
Documentation :
Submitted: 01.11.83
Revised: 18.12.93

JHITS data structure

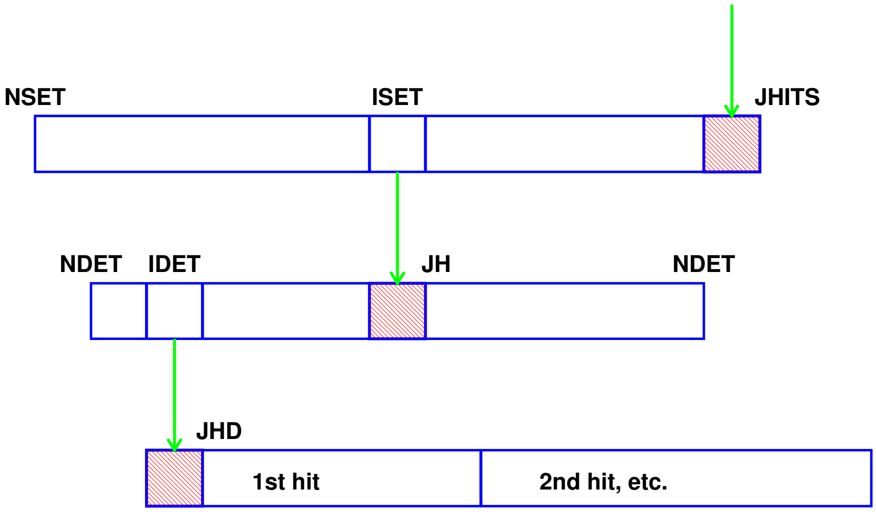


Figure 32: Layout of the JHITS data structure

JH=LQ(JHITS-ISET)	pointer to hits structure for set number ISET
IQ(JH+IDET)	number of words used for storing the hits of detector number IDET
JHD=LQ(JH-IDET)	pointer to hits bank for detector number IDET of set number ISET
IQ(JHD+1)	1^{st} word of 1^{st} hit
IQ(JHD+NWH+1)	1^{st} word of 2^{nd} hit
JS=LQ(JSET-ISET)	pointer to the structure containing the description of set number ISET
JD=LQ(JS-IDET)	pointer to the bank containing the description of detector number IDET of set number ISET
NWH=IQ(JD+3)	number of words in which a hit of detector number IDET of set number ISET is stored

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.

Origin : R.Brun, W.Gebel **Submitted:** 10.08.84
Revision : **Revised:** 18.12.93
Documentation :

Routines to store and retrieve DIGItisations

```
CALL GSDIGI (ISET,IDET,LTRA,NTRA,NUMBV,KDIGI,IDIG*)
```

ISET (INTEGER) set number;
IDET (INTEGER) detector number;
LTRA (INTEGER) array of NTRA track numbers producing this digitisation;
NUMBV (INTEGER) volume numbers corresponding to list NAMESV of GSDET;
KDIGI (INTEGER) array of current digisation elements;
IDIG (INTEGER) stored digitisation number, if 0, digitisation has not been stored.

Stores element values for current digitisation into the data structure JDIGI.

```
CALL GPDIGI (CHSET,CHDET)
```

CHSET (CHARACTER*4) set name, if '*' prints all JDIGI banks of all sets;
CHDET (CHARACTER*4) detector name, if '*' prints digitisations in all detectors of set; CHSET

Prints JDIGI banks for detector CHDET of set CHSET.

```
CALL GFDIGI (CHSET,CHDET,NTDIM,NVDIM,NDDIM,NDMAX,NUMVS,
             LTRA*,NTRA*,NUMBV*,KDIGI*,NDIGI*)
```

CHSET (INTEGER) set name;
CHDET (INTEGER) detector name;
NTDIM (INTEGER) 1^st dimension of LTRA, maxmum number of tracks contributing to each hit to be returned;
NVDIM (INTEGER) 1^{st} dimension of arrays NUMVS, NUMBV, same as argument NV of GSDET;
NDDIM (INTEGER) 1^{st} dimension of KDIGI, same as argument ND of GSDETD;
NDMAX (INTEGER) maximum number of digitisations to be returned, second dimension of arrays NTDIM, NUMBV and KDIGI;
NUMVS (INTEGER) is a 1-dim array of length NVDIM that contains the copy numbers identifying the detector to be selected, all 0 is interpreted as all copies of detector CHDET;
LTRA (INTEGER) is a 2-dim array NTDIM,NDMAX that contains for each digitisation the numbers of the tracks which have produced it;
NTRA (INTEGER) is a 1-dim array of length NDMAX that contains, for each digitisation, the number of tracks contributing, in case this number is greater than NTDIM, only the first NTDIM corresponding tracks are returned on LTRA;
NUMBV (INTEGER) is a 2-dim array NVDIM,NDMAX that contains, for each digitisation, the list of volume numbers which identify each detector;
KDIGI (INTEGER) is a 2-Dim array NDDIM,NDMAX that contains the NDIGI digitisations returned;
NDIGI (INTEGER) 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.

Returns the digitisations for the detector CHDET identified by the list of copy numbers NUMVS belonging to set CHSET. The maning of the variables is the following:

`KDIGI(1,I)`	digitisation element 1 for digitisation number `I`
`NUMBV(1,I)`	first volume number for digitisation number `I`
`LTRA (1,I)`	number of the first track 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)
```

Origin	: R.Brun	**Submitted:** 01.11.83
Revision	:	**Revised:** 18.12.93
Documentation	:	

The JDIGI data structure

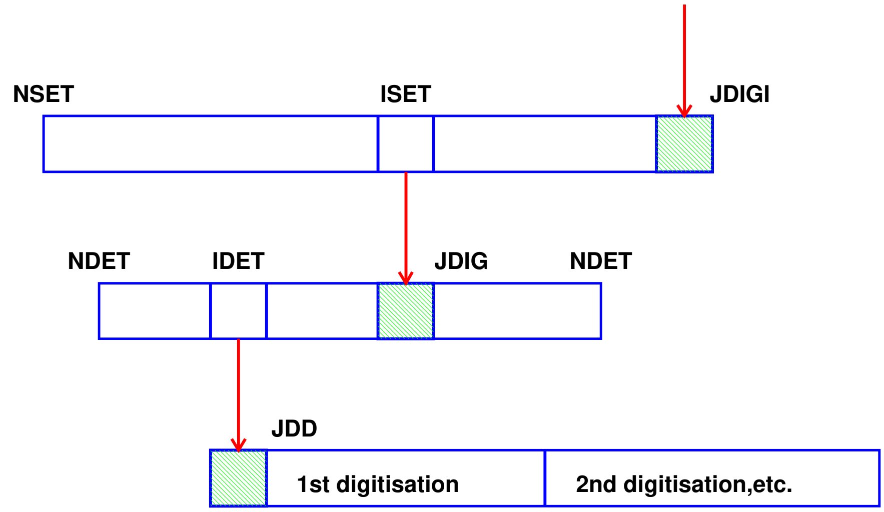


Figure 33: Layout of the JDIGI data structure

JDIG=LQ(JDIGI-ISET)	pointer to digitisations structure for set number ISET
JDD=LQ(JDIG-IDET)	pointer to digitisations of detector number IDET of set number ISET
IQ(JDIG+IDET)	pointer to last word of last digitisation for detector number IDET
IQ(JDD+1)	1^{st} word of 1^{st} digitisation
IQ(JDD+N+1)	1^{st} word of 2^{nd} digitisation
JS=LQ(JSET-ISET)	pointer to the structure containing the description of digitisations of set number ISET
JD=LQ(JS-IDET)	pointer to the structure containing the description of digitisations of detector number IDET
NWD=IQ(JD+5)	number of elements of digitisation of set/detector ISET/IDET
NTRA	number of tracks contributing to 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.

Origin : R.Brun,H. Boerner **Submitted:** 01.10.81
Revision : **Revised:** 18.12.93
Documentation :

Intersection of a track with a cylinder or a plane

```
CALL GICYL (R,X1,X2,S1,S2,IC,XINT*,SINT*,PZINT*,IFLAG*)
```

R (REAL) radius of cylinder in cm;
X1 (REAL) array of 6, (x, y, z, dx/ds, dy/ds, dz/ds) of 1^{st} point;
X2 (REAL) array of 6, (x, y, z, dx/ds, dy/ds, dz/ds) of 2^{nd} point;
S1 (REAL) track length s at 1^{st} point;
S2 (REAL) track length s at 2^{nd} point;
IC (INTEGER) type of interpolation:

- 1 straight line defined as (Xi(1),Xi(2),Xi(3)) + s (Xi(4),Xi(5),Xi(6)), where i is 1 or 2 according to which of the two points is *inside* the cylinder;
- 2 straight line going from (X1(1),X1(2),X1(3)) to (X2(1),X2(2),X2(3));
- 3 third degree curve with:

$$\vec{P}(s) = \vec{a}\, s^3 + \vec{b}\, s^2 + \vec{c}\, s + \vec{d} \qquad \mathtt{X}i = \left(\vec{P}(\mathtt{S}i),\ \left.\frac{d\vec{P}}{ds}\right|_{s=\mathtt{S}i}\right) \qquad i = 1, 2$$

XINT (REAL) array of 6 x, y, z, dx/ds, dy/ds, dz/ds at intersection point;
SINT (REAL) S at intersection point;
PZINT (REAL) ϕ, z, $d\phi/dr$, dz/dr in cylindrical coordinates at intersection point;
IFLAG (INTEGER) return flag:

- 0 track does not intersect cylinder;
- 1 track intersects cylinder.

Calculates intersection of track with a cylinder of radius R. The track is approximated by a cubic in the track length. To improve stability, the coordinate system is shifted.

```
CALL GIPLAN (YC,X1,X2,S1,S2,IC,XINT*,SINT*,PZINT*,IFLAG*)
```

The arguments have the same meaning than in the previous routine, apart from:

YC (REAL) y coordinate of plane;

Calculates intersection of track with a plane parallel to $x - z$. The track is approximated by a cubic in the track length. To improve stability, the coordinate system is shifted.

Note: the default accuracy is 10 microns. The value of EPSI (internal variable) must be changed for a better precision.

Origin : GEANT2
Revision :
Documentation :

Submitted: 01.10.81
Revised: 19.12.93

Digitisation for drift- or MWP- Chambers

`CALL` **GPDRIF** `(DETREP,HITREP,IOUT*)`

`DETREP` (`REAL`) array of 6 with detector description:

1 number of wires;
2 wire spacing in cm;
3 $\sin(\alpha)$;
4 $\cos(\alpha)$;
5 distance of wire 1 from the origin in cm;
6 drift velocity (cm/nsec)

`HITREP` (`REAL`) array of 2 with track description:

1 x coordinate of intersection;
2 y coordinate of intersection;

`IOUT` (`INTEGER`) result of digitisation:

1 wire number;
2 drift time (signed to avoid left/right ambiguity).

Digitisation routine for a plane drift chamber. The drift chamber is supposed to be in the $x - y$ plane. The normal to the wires in the $x - y$ plane and the z axis form an angle α.

`CALL` **GCMWPC** `(DETREP,HITREP,IOUT*)`

`DETREP` (`REAL`) array of 6 with detector description:

1 number of wires;
2 wire spacing (radians);
3 $d\theta/dz$ along the wires;
4 θ of a point on wire 1;
5 z of a point on wire 1 corresponding to θ=`DETREP(4)`;
6 gap width;

`HITREP` (`REAL`) array of 4 with track description:

1 θ coordinate of intersection;
2 z coordinate;
3 $d\theta/dr$;
4 dz/dr;

`IOUT` (`INTEGER`) array of 4 with result of digitisation:

1 wire number (-1 if missing);
2 cluster size;
3 wire number of second cluster if any;
4 cluster size;

Routine to compute one or two digitisations produced by a hit on a cylindrical `MWPC`. The chamber is a cylinder of thickness `DETREP(6)` which should be small compared with the radius.

`CALL` **GPMWPC** `(DETREP,HITREP,IOUT*)`

`DETREP` (`REAL`) array of 6 with detector description:

- `1` number of wires;
- `2` wire spacing (radians);
- `3` $\sin(\alpha)$;
- `4` $\cos(\alpha)$;
- `5` distance of wire 1 from the origin in cm;
- `6` gap width;

`HITREP` (`REAL`) array of 4 with track description:

- `1` x coordinate of intersection;
- `2` y coordinate of intersection;
- `3` dx/dz;
- `4` dy/dz;

`IOUT` (`INTEGER`) array of 2 with result of digitisation:

- `1` wire number (-1 if missing);
- `2` cluster size;

Digitisation routine for a plane `MWPC`.

Origin : W.Mitaroff **Submitted:** 21.02.85
Revision : **Revised:** 19.12.93
Documentation :

Digitisation for drift chambers

```
CALL GCDRIF (RADD,ZMIN,ZMAX,DETREP,HITREP,IOUT*)
```

`RADD` (`REAL`) radius of cylindrical chamber in cm;

`ZMIN` (`REAL`) z of lower end of cylindrical chamber;

`ZMAX` (`REAL`) z of upper end of cylindrical chamber;

`DETREP` (`REAL`) array of 8 with detector description:

- 1 number of wires;
- 2 wire spacing in ϕ (radians);
- 3 cosine of wire angle with respect to the z axis;
- 4 sine of wire angle with respect to the z axis (signed like $d\phi/dz$);
- 5 $d\phi/dz$ along wire;
- 6 ϕ of point with $z = 0$ on wire 1;
- 7 drift velocity (cm nsec^{-1});
- 8 if > 0 user routine `GUDTIM` will be called to calculate drift time;

`HITREP` (`REAL`) array of 4 describing the track:

- 1 ϕ coordinate of intersection;
- 2 z coordinate of intersection;
- 3 $d\phi/dr$;
- 4 dz/dr;

`IOUT` (`INTEGER`) array of 4 with digitisation information:

- 1 wire number (1...`NWI` with increasing phi), -1 if `DETREP` parameters are inconsistent;
- 2 drift time in nsec, > 0 if $\phi(hit) > \phi(wire)$;
- 3 digitised current division information (relative position of charge along wire, per mille);
- 4 amount of charge deposited onto wire.

Digitisation routine for a cylindrical drift chamber.

```
                  Charge
   .              |                        .
   |              .                        |
   =========================================  SENSE WIRE
...................................................> Z (cm)
   Z              Z                        Z
    l                                       u
...............................................> ICD (0<ICD<1000)
   0              ICD                      1000
         ICD                 (1000-ICD)
```

Figure 34: Coordinate system along the wire

Knowing the position Z of the deposit of charge we can calculate

$$\texttt{ICD} = L\frac{Z - Z_l}{Z_u - Z_l}$$

where $L = 1000$ in the program. This is the information stored into IOUT(3).

CALL **GCDERR** (ICD*,ERP,ERS)

ICD (INTEGER) digitised current division information ($\leq$ ICD ≤ 1000), overwritten on output with the modified value taking into account the errors;

ERP (REAL) variance of Gaussian pedestal errors on the measured pulse heights relative to the sum of the pulse heights;

ERS (REAL) variance of Gaussian slope errors on the measured pulse heights relative to the each pulse heights.

Routine to calculate the error on the current division information as obtained by GCDRIF. Here we assume that ICD has been determined by measuring the pulse heights I_1, I_2 at the two ends of the wire with the formula:

$$\texttt{ICD} = L\,\frac{I_2}{I_+} \qquad \text{with} \qquad I_+ = I_1 + I_2$$

Its error is determined by:

$$\delta\texttt{ICD} = -\frac{\texttt{ICD}}{I_+}\delta I_1 + \frac{L - \texttt{ICD}}{I_+}\delta I_2 \quad \text{and} \quad \delta I_1 = \delta_1 + \epsilon_1\, I_1 \;\; ; \;\; \delta I_2 = \delta_2 + \epsilon_2\, I_2$$

δ_1 and δ_2 are of dimension [I] and represent the *pedestal* errors. ϵ_1 and ϵ_2 are the *slope* errors. Errors are independent (no correlations), with a Gaussian distribution with average 0 and ERP as relative variance for pedestals δ_i/I_+ and ERS as variance for slopes ϵ_i. This gives the final result

$$\delta\texttt{ICD} = \underbrace{-\frac{\delta_1}{I_+}\texttt{ICD} + \frac{\delta_2}{I_+}(L - \texttt{ICD})}_{\textit{pedestals}} + \underbrace{(\epsilon_2 - \epsilon_1)\frac{\texttt{ICD}(L - \texttt{ICD})}{L}}_{\textit{slope}}$$

GCDERR sets the ICD obtained from GCDRIF to ICD $=$ ICD $+\ \delta$ICD with $0 \leq$ ICD $\leq L$.

VALUE = **GUDTIM** (DETREP,HITREP,IW1,DIS)

The arguments have the same meaning than for GCDRIF apart from:

IW1 (INTEGER) wire number which will generate a signal;

DIS (REAL) distance from the track to the wire;

This function has to be written by the user to return the drift time in nanoseconds.

HITS Bibliography

Origin :
Revision :
Documentation : R.Brun

Submitted: 15.08.84
Revised: 16.12.93

The I/O routines

The I/O routines allow one to read, write or print the GEANT data structures. All GEANT I/O is handled though the ZEBRA system, which provides the following facilities:

- native and machine independent I/O to and from disk and tape devices;
- sequential and direct access I/O (from disk);

This allows, for example, to generate events on one machine, dump the data structures to a mass storage device, and then read them back for further processing on a different machine.

This section describes the various routines for I/O in GEANT. For a complete reference on the ZEBRA I/O system the user is referred to the ZEBRA documentation [1, 8].

Origin : R.Brun
Revision :
Documentation :
Submitted: 01.06.83
Revised: 16.12.93

ZEBRA sequential files handling

```
CALL GOPEN (LUN,CHOPT,LEN,IER*)
```

LUN (INTEGER) logical unit number, 1≤LUN<100;

CHOPT (CHARACTER*(*)) any valid combination of FZFILE options (see [1, 8]), the most common being:

- I input file;
- O output file;
- X exchange mode file;

LEN (INTEGER) maximum record length, for more details see routine FZFILE in the ZEBRA manual;

IER (INTEGER) error flag:

- 0 legal open request;
- 1 attempt to open more than 5 ZEBRA sequential files;

Opens a ZEBRA FZ file for sequential I/O of GEANT data structures. No more than 5 files can be open at any one time.

```
CALL GCLOSE (LUN,IER*)
```

LUN (INTEGER) logical unit to be closed, if 0 all ZEBRA FZ streams will be closed;

IER (INTEGER) error flag:

- 0 legal close request;
- 1 no unit was found open;

This routine closes one or all ZEBRA FZ file(s). This is a mandatory routines for all open ZEBRA files before the end of the program. Failure to call this routine may cause files to be incompletely written to the storage media.

Origin : R. Brun **Submitted:** 01.06.83
Revision : **Revised:** 17.12.93
Documentation :

Data structure I/O with sequential files

```
CALL GFIN (LUN,CHOBJ,NKEYS,IDVERS*,CHOPT,IER*)
```

LUN (INTEGER) logical unit;

CHOBJ (CHARACTER*4) array with the data structures to be read:

- MATE material;
- TMED tracking medium;
- VOLU volumes;
- ROTM rotation matrix;
- SETS detector set;
- PART particle;
- SCAN scan geometry;
- DRAW drawing;
- INIT all of the above;
- KINE this keyword will trigger the read of KINE and VERT unless the flag S is set;
- DIGI digitisation;
- HEAD event header;
- HITS hits;
- RUNG run;
- STAK particle temporary stack;
- STAT volume statistic;
- VERT vertex;
- JXYZ track points;
- TRIG this keyword will trigger the read of DIGI, HEAD, HITS, KINE, VERT and JXYZ unless the S flag is set;

NKEYS (INTEGER) number of keys in array CHOBJ;

IDVERS (INTEGER) version of the data structure to be read in, if 0 it will read any version, on output it contains the version of the data structure read in;

CHOPT (CHARACTER*(*)) option:

- I among the data structures specified in CHOBJ read only the initialisation ones;
- K among the data structures specified in CHOBJ read only KINE and TRIG, if present;
- T among the data structures specified in CHOBJ read only DIGI, HEAD, HITS, KINE, VERT and JXYZ, if present;
- S interpret KINE to mean only the KINE data structure and ignore TRIG and INIT;
- Q quiet option, no message is printed;

IER (INTEGER) error flag:

- <0 ZEBRA error flag as returned in IQUEST(1);
- 0 read completed successfully;
- >0 if only IER] structures read in out of the NKEYS requested;

Routine to read GEANT object(s) from a ZEBRA/FZ file. The data structures from disk are read in memory. The FZ data base must have been been created via GOPEN/GFOUT. Example:

```
CALL GOPEN(1,'I',1024,IER)
IDVOLU = 0
IDMATE = 0
IDTMED = 0
IDROTM = 0
IDPART = 0
IDSCAN = 0
IDSETS = 0
CALL GFIN (1,'VOLU',1,IDVOLU,' ',IER)
CALL GFIN (1,'MATE',1,IDMATE,' ',IER)
CALL GFIN (1,'TMED',1,IDTMED,' ',IER)
CALL GFIN (1,'ROTM',1,IDROTM,' ',IER)
CALL GFIN (1,'PART',1,IDPART,' ',IER)
CALL GFIN (1,'SCAN',1,IDSCAN,' ',IER)
CALL GFIN (1,'SETS',1,IDSETS,' ',IER)
```

CALL **GFOUT** (LUN,CHOBJ,NKEYS,IDVERS,CHOPT,IER*)

The arguments have the same meaning than for GFIN with the following differences:

IDVERS (INTEGER) version identifier of the data structures to be written out;

Routine to write GEANT object(s) into a ZEBRA/FZ file. The data structures from memory are written on the file. The FZ data base must have been been created via GOPEN/GFOUT. Example:

```
CALL GOPEN(1,'O',1024,IER)
CALL GFOUT (1,'VOLU',1,IDVOLU,' ',IER)
CALL GFOUT (1,'MATE',1,IDMATE,' ',IER)
CALL GFOUT (1,'TMED',1,IDTMED,' ',IER)
CALL GFOUT (1,'ROTM',1,IDROTM,' ',IER)
CALL GFOUT (1,'PART',1,IDPART,' ',IER)
CALL GFOUT (1,'SCAN',1,IDSCAN,' ',IER)
CALL GFOUT (1,'SETS',1,IDSETS,' ',IER)
```

Note: the routines GGET and GSAVE are obsolete and should not be used.

Origin : R.Brun **Submitted:** 20.08.87
Revision : **Revised:** 17.12.93
Documentation :

ZEBRA direct access files handling

I/O with direct access files is the most efficient way to use direct access devices. All the routines described in this section and in ([IOPA500]) are based on the ZEBRA-RZ package and the user is referred to the related documentation [8] for an understanding of the basic principles.

```
CALL GRFILE (LUN,CHFILE,CHOPT)
```

LUN (INTEGER) logical unit number;
CHFILE (CHARACTER*(*)) file name;
CHOPT (CHARACTER*(*)) any valid option for RZOPEN, RZMAKE and RZFILE and in particular:
- N create a new file;
- U open an existing file for update;
- Q the initial allocation (default 1000 records) is in IQUEST(10);
- I read all initialisation data structures (see description of GRIN/GROUT) from file to memory;
- O write all initialisation data structures from memory to file;

This routine opens a file for direct access I/O of ZEBRA data structures. By default the file is opened in exchange mode and the option W is added, which creates an RZ top level directory with the name //LUNnn where nn is the logical unit number.

The default allocation of an RZ directory is enough for 1000 records, which allows for a 4Mb file with the standard record length of 1024 words. If a larger file is needed, the size of the directory should be changed when creating the file. This can be done by assigning to the variable IQUEST(10) in the common /QUEST/ the requested number of records, and calling GRFILE with the option Q as shown in the following example:

```
COMMON / QUEST / IQUEST(100)
.
.
.
IQUEST(10) = 5000
CALL GRFILE(1,'mygeom.geom','Q')
```

```
CALL GRMDIR (CHDIR,CHOPT)
```

CHDIR (CHARACTER*(*)) subdirectory name to create;
CHOPT option string:
- ' ' create a subdirectory;
- S create a subdirectory and set the current directory to it;

This routine creates an RZ subdirectory.

```
CALL GREND (LUN)
```

LUN (INTEGER) logical unit number (see RZEND in the ZEBRA manual [8]);

Routine to close an RZ file. It is very important to call this routine for all RZ files open in write mode. Failure to do may result in the files being incompletely written or corrupted.

Origin : R.Brun **Submitted:** 20.08.87
Revision : **Revised:** 17.12.93
Documentation :

Data structure I/O with direct access files

CALL **GRIN** (CHOBJ,IDVERS*,CHOPT)

CHOBJ (CHARACTER*(4)) name of the data structure to read in, see [IOPA300] for more information on this;

IDVERS (INTEGER) version of the data structure to be read in, if 0 it will read any version, on output it contains the version of the data structure read in;

CHOPT (CHARACTER*(*)) option:

- I perform read only if CHOBJ is an initialisation data structure;
- K perform read only if CHOBJ is KINE or TRIG;
- T perform read only if CHOBJ is DIGI, HEAD, HITS, KINE, VERT or JXYZ;
- Q quiet option, no message is printed;

This routine reads GEANT data structures from the current working directory of an RZ file (see the ZEBRA documentation for a clarification of this concept). Example:

```
CALL GRFILE(1,'Geometry.dat',' ')
CALL GRIN ('VOLU',1,' ')
CALL GRIN ('MATE',1,' ')
CALL GRIN ('TMED',1,' ')
CALL GRIN ('ROTM',1,' ')
CALL GRIN ('PART',1,' ')
CALL GRIN ('SCAN',1,' ')
CALL GRIN ('SETS',1,' ')
```

The same result can be achieved by:

```
CALL GRFILE(1,'Geometry.dat','I')
```

CALL **GROUT** (CHOBJ,IDVERS,CHOPT)

The meaning of the arguments is the same than in the previous routine GRIN, but for writing instead than for reading.

This routine writes GEANT data structures into the current working directory of an RZ file (see the ZEBRA documentation for a clarification of this concept). Note that if the cross-sections and energy loss tables are available in the data structure JMATE, then they are saved on the data base. The data structures saved by this routine can be retrieved with the routine GRIN. Before calling this routine a
tt RZ data base must have been created using GRFILE. The data base must be closed with RZEND. Example:

```
CALL GRFILE(1,'Geometry.dat','N')
CALL GROUT ('VOLU',1,' ')
CALL GROUT ('MATE',1,' ')
CALL GROUT ('TMED',1,' ')
```

```
CALL GROUT ('ROTM',1,' ')
CALL GROUT ('PART',1,' ')
CALL GROUT ('SCAN',1,' ')
CALL GROUT ('SETS',1,' ')
```

The same result can be achieved by:

```
CALL GRFILE(1,'Geometry.dat','NO')
```

The interactive version of tt GEANT provides facilities to interactively update, create and display objects. The routines `GRGET` and `GRSAVE` are obsolete and should not be used.

IOPA Bibliography

[1] J.Zoll. *ZEBRA - Reference Manual - FZ Sequential I/O,* Program Library Q100. CERN, 1991.

[2] R.Brun, M.Goossens, and J.Zoll. *ZEBRA Users Guide,* Program Library Q100. CERN, 1991.

Origin :
Revision :
Documentation :

Submitted: 28.04.86
Revised: 15.12.93

Section KINE

This section describes the GEANT kinematics structures JVERTX and JKINE and the routines which permit one to store and retrieve the information. It also contains a short description of the GEANT interface to the JETSET generator and related routines.

Origin : R.Brun
Revision : M.Maire
Documentation : R.Brun

Submitted: 01.06.83
Revised: 14.12.93

Storing and retrieving vertex and track parameters

`CALL` **GSVERT** `(VERT,NTBEAM,NTTARG,UBUF,NUBUF,NVTX*)`

`VERT` (`REAL`) array of 3 containing the position of the vertex in the `MRS`;
`NTBEAM` (`INTEGER`) beam track number origin of the vertex (0 if none exists);
`NTTARG` (`INTEGER`) target track number origin of the vertex (0 if none exists);
`UBUF` (`REAL`) user array of `NUBUF` user words to be stored in the data structure together with the vertex;
`NWBUF` (`INTEGER`) number of user words;
`NVTX` (`INTEGER`) number of the vertex stored (0 in case of error).

This routine stores the parameters of a vertex to which tracks should be attached via calls to `GSKINE`. Vertices are stored in the `JVERTX` data structure (see `[KINE199]`) and they are preserved for the whole duration of the event.

`GEANT` tracks one vertex at a time: all the particles of one vertex, and all the secondaries generated thereof must be transported before the particles of the next vertex are considered.

This routine should be called only from `GUKINE` when the initial kinematics of the event is defined. During tracking it is possible to add vertices to the `JVERTX` data structure, but this should be done via the `GSKING` routine and the appropriate variables in the `/GCKING/` common block. Please see `[TRAK300]` for more information.

`CALL` **GSVERU** `(NVTX,NUBUF,UBUF,IADR*)`

`NVTX` (`INTEGER`) vertex number;
`NUBUF` (`INTEGER`) number of user words;
`UBUF` (`REAL`) array of user words;
`IADR` (`INTEGER`) position where information is stored in the user bank of the vertex.

This routine stores the first `NUBUF` words of array `BUF` in the user bank attached to vertex `NVTX`, starting at location `IADR+1`. On exit `IADR` is set to `IADR+NUBUF`, allowing subsequent filling. This allows effectively to *add* information to an existing vertex, whether or not it has already an user buffer. This is the only way to add user information to a vertex which has been created at tracking time by the routine `GSKING` via an appropriate setting of parameters in the common `/GCKING/`.

`CALL` **GFVERT** `(NVTX,VERT*,NTBEAM*,NTTARG*,TOFG*,UBUF*,NUBUF*)`

Retrieves vertex parameters. The parameters have the same meaning than the ones of `GSVERT`.

`TOFG` (`REAL`) time of flight when the vertex was created;

`CALL` **GPVERT** `(IVTX)`

Prints vertex parameters.

`IVTX` (`INTEGER`) vertex number to be printed, all vertices if IVTX=0;

`CALL` **GSKINE** `(PLAB,IPART,NV,UBUF,NUBUF,NT*)`

`PLAB` (`REAL`) array of 3 containing the components of the momentum of the particle in GeV c^{-1};

IPART	(INTEGER) particle code (see [CONS300]);
NV	(INTEGER) number of the vertex origin of track;
UBUF	(REAL) array of NUBUF user words to be stored with the track;
NUBUF	(INTEGER) number of user words;
NT	(INTEGER) number of the stored track (0 in case of error).

This routine stores the parameters of a track in the JKINE data structure (see [KINE199]). This data structure survives for the whole event, and it is wiped out by GTRIGI at the beginning of each new event. There are two reasons to store tracks in the JKINE data structure:

- at the beginning of an event, from within GUKINE, to define the initial kinematics of the event, i.e. the particles which GEANT will transport. This must be done through GSKINE after that the vertices from which the particles originate have been defined through GSVERT;
- during tracking, to preserve the kinematic information of a particular track till the end of the current event. In this case the user should not call GSKINE directly, but rather set the appropriate flags in the /GCKING/ common block and call GSKING. This is explained in detail in [TRAK300].

```
CALL GSKINU (NT,NUBUF,UBUF,IADR*)
```

NT	(INTEGER) track number;
NUBUF	(INTEGER) number of user words;
UBUF	(REAL) array of user words;
IADR	(INTEGER) position where information is stored in the user bank of the track.

This routine stores the first NUBUF words of array BUF in the user bank attached to track NT, starting at location IADR+1. On exit IADR is set to IADR+NUBUF, allowing subsequent filling. This allows effectively to *add* information to an existing track, whether or not it has already an user buffer. This is the only way to add user information to a track which has been created at tracking time by the routine GSKING via an appropriate setting of parameters in the common /GCKING/.

```
CALL GFKINE (ITRA,VERT*,PVERT*,IPART*,NVERT*,UBUF*,NUBUF*)
```

Retrieves long-life track parameters. The parameters have the same meaning than the ones of GSKINE.

```
CALL GPKINE (ITRA)
```

Prints initial track parameters.

ITRA	(INTEGER) track to be printed, all tracks if IVTX=0;

Origin : R.Brun
Revision :
Documentation :

Submitted: 01.11.83
Revised: 14.12.93

The data structures JVERTX and JKINE

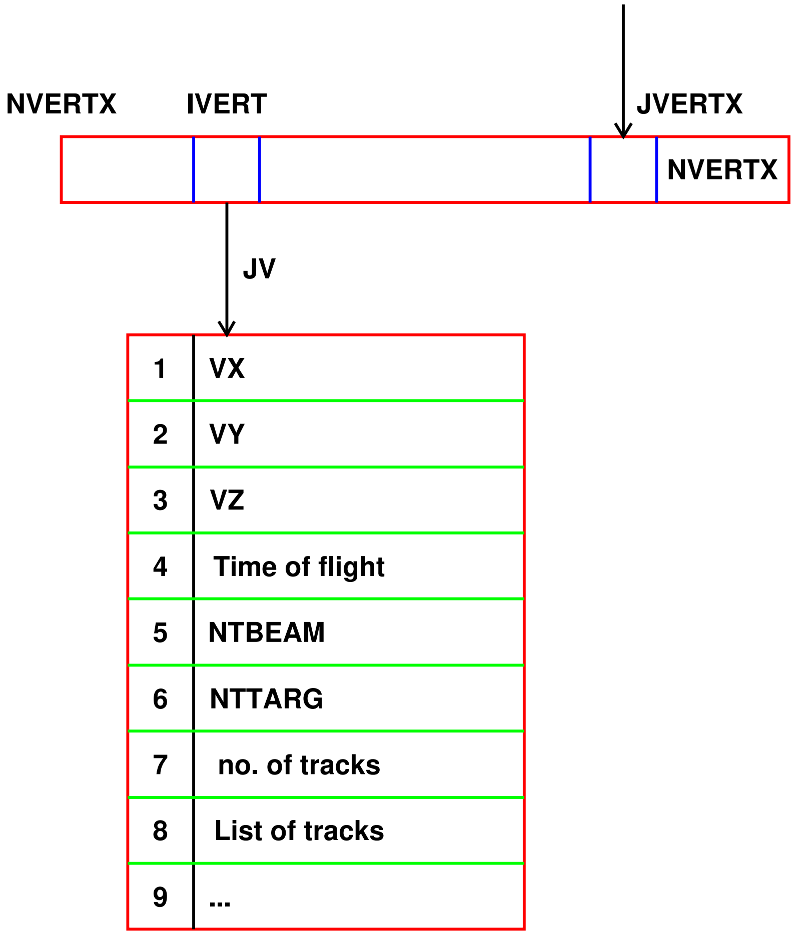


Figure 35: Layout of the JVERTX data structure

JV = LQ(JVERTX-IVERT) pointer to the bank of vertex IVERT.
JVU = LQ(JV-1) pointer to the user part of the bank.

The JVERTX banks are filled by the routine GSVERT and GSVERU. Vertex parameters can be retrieved by the routine GFVERT and printed by the routine GPVERT.

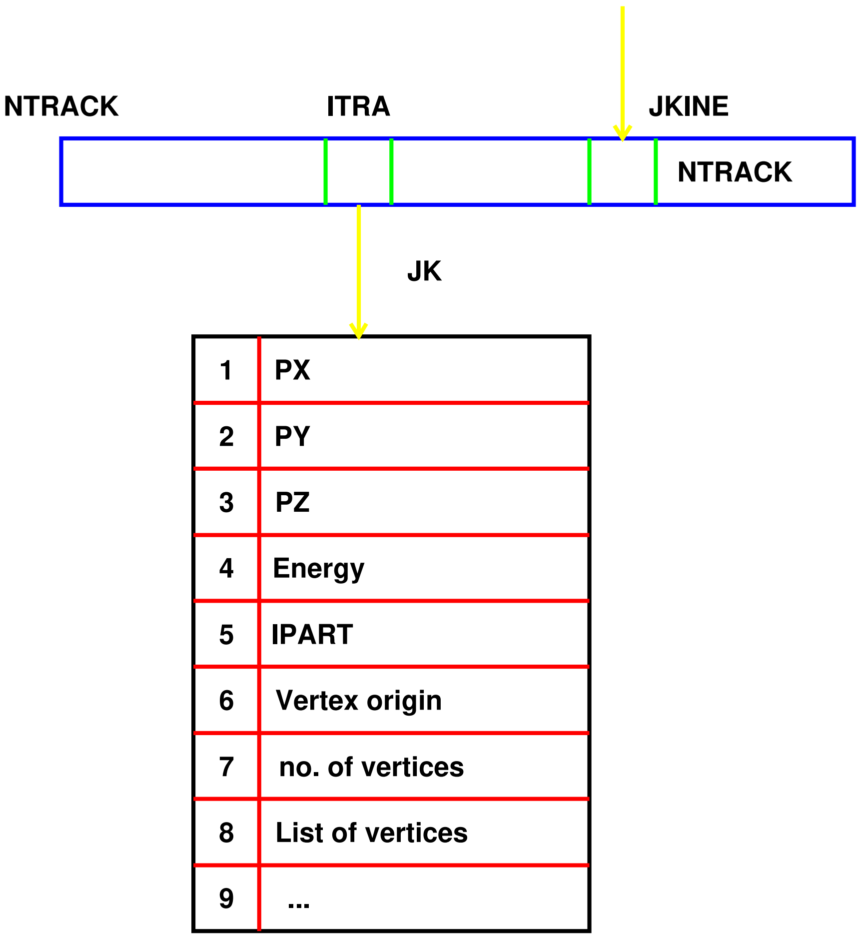


Figure 36: Layout of the JKINE data structure

JK = LQ(JKINE-ITRA) pointer to the bank of track ITRA.
JKU = LQ(JK-1) pointer to the user part of the bank.

The JKINE banks are filled by the routine GSKINE and GSKINU. Track parameters can be retrieved by the routine GFKINE and printed by the routine GPKINE.

Origin : R.Brun,R.Hemingway
Revision :
Documentation :

Submitted: 01.11.83
Revised: 15.12.93

Interface to the Lund Monte Carlo

GEANT contains a simple interface to the JETSET Monte Carlo generator [1]. This interface is composed by an initialisation routine and an event generation routine.

CALL **GLUNDI**

This is the initialisation routine which should be called before the call to GFFGO, as it defines the data record LUND. This routine performs the following actions:

1. sets ECLUND= $\sqrt{s} = 92.25$ GeV;
2. sets IFLUND= 0;
3. activates initial state radiative effects MSTJ(107)= 1;
4. declares some particles (K_s^0, $\Sigma^\pm$, $\Xi^\pm$, Λ^0, Ω^-) to be stable in JETSET so that GEANT can handle their tracking and decays:

```
MDCY(LUCOMP(310) ,1)=0
MDCY(LUCOMP(3222),1)=0
MDCY(LUCOMP(3112),1)=0
MDCY(LUCOMP(3322),1)=0
MDCY(LUCOMP(3312),1)=0
MDCY(LUCOMP(3122),1)=0
MDCY(LUCOMP(3334),1)=0
```

Variables ECLUND, total energy in the CMS, and IFLUND, quark flavours produced, are used as arguments for LUEEVT called by GLUND and they can be changed via the LUND data record (see [BASE040]).

CALL **GLUND**

Generate a $e^- e^+$ collision via the JETSET MonteCarlo and stores the resulting final state particles for transport by GEANT. All particles are attached to a vertex in (0.,0.,0.). The K and P arrays containing the full JETSET event are copied in the user buffer of the generated vertex, with the K variables converted into REAL numbers. The content of the user buffer is the following:

UBUF(1)	527.0	
UBUF(2)	N number of particles	
UBUF(3)	K(1,1)	
UBUF(4)	K(1,2)	
UBUF(5)	K(1,3)	
UBUF(6)	K(1,4)	
UBUF(7)	K(1,5)	N times
UBUF(8)	P(1,1)	
UBUF(9)	P(1,2)	
UBUF(10)	P(1,3)	
UBUF(11)	P(1,4)	
UBUF(12)	P(1,5)	

Origin : R.Hemingway
Revision :
Documentation :

Submitted: 01.11.83
Revised: 15.12.93

$\tau^{\pm}$ generation and decay

CALL **GALL GTAU**

This routine is now dummy and it will only print an error message. The appropriate routines from a generator Monte Carlo (for instance the JETSET [1] package) should be used to handle $\tau^{\pm}$ production and decay.

KINE Bibliography

[1] T.Sjöstrand et al. *The LUND MonteCarlo Programs*, CERN Program Library Pool Writeup W5035 / W5045 / W5046 / W5047 / W5048 edition, November 1989.

Origin :
Revision :
Documentation : M. Maire, F.Carminati

Submitted: 24.09.84
Revised: 26.07.93

Introduction to the section PHYS

1 Summary of the physics processes

The computer simulation of particles traversing an experimental setup has to take into account the interactions of those particles with the material of the detector. `GEANT` is able to simulate the dominant processes which can occur in the energy range **from 10 keV to 10 TeV** for electromagnetic interactions. As far as hadronic interactions are concerned, the range of validity is the one of the hadronic packages used, which usually does not extend below a few tens of MeV. For more information the user is invited to consult the relevant documentation [1, 2, 3, 4, 5, 6].

Simulating a given process means:

- Evaluating the probability of occurrence of the process, by sampling the **total cross-section** of the process.
- Generating the final state after interaction, by sampling the **differential cross-section** of the process.
- In case of (quasi-)continuous processes, e.g. CSDA (Continuous Slowing Down Approximation), energy losses or multiple scattering, computing the mean values of some characteristic quantities.

In Table 1 below we summarise all the processes currently implemented in `GEANT`, with a reference to the corresponding sections.

1.1 Simulated Processes

Hadronic Interactions

To simulate the interactions of hadrons with the nuclei of the matter traversed, two alternatives are provided:

1. The generator of the `FLUKA` [2, 3] hadron shower MonteCarlo and the interface routines to `GEANT`. See `[PHYS520]` for more information.

2. The generator of the `GHEISHA [1]` hadron shower MonteCarlo and the interface routines to `GEANT`. See `[PHYS510]` for more information.

The code both of the `GHEISHA` and of the `FLUKA` generators is contained in the `GEANT` library. Users should be aware that the routines of these packages do not follow the `GEANT` naming conventions and therefore they can clash with the names of user procedures.

Electromagnetic Processes

By means of systematic fits to the existing data, the cross-sections of the electromagnetic processes are well reproduced (within a few percent) from 10 keV up to 100 GeV, both for light (low Z) and for heavy materials.

This feature, together with the use of the interface with one of the hadronic shower generators available, makes `GEANT` useful also for shower simulation in gases.

Muonic interactions

Muonic interactions are simulated up to 10 TeV, making `GEANT` useful for cosmic rays studies.

Ionisation by charged particles

The following alternatives are provided to simulate this process:

- Sampling from the appropriate distribution around the mean value of the energy loss ([PHYS332]).
- Explicit generation of δ-rays (see [PHYS330/331/430]) and restricted fluctuations below the energy threshold for the production of δ-rays.
- Sampling the contribution of single collisions from statistical distributions. This can be used as an alternative to the first one when simulating energy losses in very thin layers (small value of g cm^{-2}) (see [PHYS334]).

Full Landau fluctuations and generation of δ-rays cannot be used together in order to avoid double counting of the fluctuations. An automatic protection has been introduced in GEANT to this effect. See [PHYS333/332] and [BASE040] for further information.

Multiple Scattering

Two methods are provided:

1. Molière distribution or plural scattering ([PHYS325], [PHYS328]).
2. Gaussian approximation ([PHYS320]).

The JMATE data structure

In order to save time during the transport of the particles, relevant energy-dependent quantities are tabulated at the beginning of the run, for all materials, as functions of the kinetic energy of the particle. In particular, the inverse of the total cross-sections of all processes involving photons, electrons and muons and the dE/dx and range tables for electrons, muons and protons are calculated. The actual value of, say, the interaction length for a given process (i.e. the inverse of the macroscopic cross section) is then obtained via a linear interpolation in the tables. The data structure which contains all this information in memory is supported by the link JMATE in the /GCLINK/ common block. See [PHYS100] and [CONS199] for a more information on these tables.

Probability of Interaction

The total cross-section of each process is used at tracking time to evaluate the probability of occurrence of the process. See [PHYS010] for an explanation of the method used.

Note: The section PHYS is closely related to the section CONS. Users wishing to have a complete overview of the physics processes included in GEANT should read both sections.

1.2 Control of the physical processes

For most of the individual processes the default option (indicated below) can be changed via data records [BASE040]. The processes are controlled via a control variable which is in the common /GCKING/. If not otherwise noted, the meaning of the control variable is the following:

= 0 The process is completely ignored.

= 1 The process is considered and possible secondary particles generating from the interaction are put into the /GCKING/ common. If the interacting particle disappears in the interaction, then it is stopped with ISTOP=1 (common /GCTRAK/)

= 2 The process is considered. If secondary particles result from the interaction, they are not generated and their energy is simply added in the variable DESTEP (common /GCTRAK/. If the interacting particle disappears in the interaction, the variable ISTOP is set to 2.

Below are listed the data record keywords, the flag names and values, and the resulting action:

Keyword Related process

`DCAY` Decay in flight. The decaying particles stops. The variable `IDCAY` controls this process.

- `IDCAY =0` No decay in flight.
- `=1` (**D**) Decay in flight with generation of secondaries.
- `=2` Decay in flight without generation of secondaries.

`MULS` Multiple scattering. The variable `IMULS` controls this process.

- `IMULS =0` No multiple scattering.
- `=1` (**D**) Multiple scattering according to Molière theory.
- `=2` Same as `1`. Kept for backward compatibility.
- `=3` Pure Gaussian scattering according to the Rossi formula.

`PFIS` Nuclear fission induced by a photon. The photon stops. The variable `IPFIS` controls this process.

- `IPFIS =0` (**D**) No photo-fission.
- `=1` Photo-fission with generation of secondaries.
- `=2` Photo-fission without generation of secondaries.

`MUNU` Muon-nucleus interactions. The muon is not stopped. The variable `IMUNU` controls this process.

- `IMUNU =0` No muon-nucleus interactions.
- `=1` (**D**) Muon-nucleus interactions with generation of secondaries.
- `=2` Muon-nucleus interactions without generation of secondaries.

`LOSS` Continuous energy loss. The variable `ILOSS` controls this process.

- `ILOSS =0` No continuous energy loss,IDRAY is forced to 0.
- `=1` Continuous energy loss with generation of δ-rays above `DCUTE` (common `/GCUTS/`) and restricted Landau fluctuations below `DCUTE`.
- `=2` (**D**) Continuous energy loss without generation of δ-rays and full Landau-Vavilov-Gauss fluctuations. In this case the variable `IDRAY` is forced to `0` to avoid double counting of fluctuations.
- `=3` Same as 1, kept for backward compatibility.
- `=4` Energy loss without fluctuation. The value obtained from the tables is used directly.

`PHOT` Photoelectric effect. The interacting photon is stopped. The variable `IPHOT` controls this process.

- `IPHOT =0` No photo-electric effect.
- `=1` (**D**) Photo-electric effect with generation of the electron.
- `=2` Photo-electric effect without generation of the electron.

`COMP` Compton scattering. The variable `ICOMP` controls this process.

- `ICOMP =0` No Compton scattering.
- `=1` (**D**) Compton scattering with generation of e^-.
- `=2` Compton scattering without generation of e^-.

`PAIR` Pair production. The interacting γ is stopped. The variable `IPAIR` controls this process.

- `IPAIR =0` No pair production.
- `=1` (**D**) Pair production with generation of e^-/e^+.
- `=2` Pair production without generation of e^-/e^+.

`BREM` bremsstrahlung. The interacting particle (e^-, e^+, μ^+, μ^-) is not stopped. The variable `IBREM` controls this process.

- `IBREM =0` No bremsstrahlung.
- `=1` (**D**) bremsstrahlung with generation of γ.

=2 bremsstrahlung without generation of γ.

RAYL Rayleigh effect. The interacting γ is not stopped. The variable IRAYL controls this process.

IRAYL =0 (**D**) No Rayleigh effect.
=1 Rayleigh effect.

DRAY δ-ray production. The variable IDRAY controls this process.

IDRAY =0 No δ-rays production.
=1 (**D**) δ-rays production with generation of e^-.
=2 δ-rays production without generation of e^-.

ANNI Positron annihilation. The $e^+$is stopped. The variable IANNI controls this process.

IANNI =0 No positron annihilation.
=1 (**D**) Positron annihilation with generation of photons.
=2 Positron annihilation without generation of photons.

HADR Hadronic interactions. The particle is stopped in case of inelastic interaction, while it is not stopped in case of elastic interaction. The variable IHADR controls this process.

IHADR =0 No hadronic interactions.
=1 (**D**) Hadronic interactions with generation of secondaries.
=2 Hadronic interactions without generation of secondaries.
>2 Can be used in the user code GUPHAD and GUHADR to chose a hadronic package. These values have no effect on the hadronic packages themselves.

LABS Light ABSorption. This process is the absorption of light photons (particle type 7) in dielectric materials. It is turned on by default when the generation of Čerenkov light is requested (data record CKOV). For more information see [PHYS260].

ILABS =0 No absorption of photons.
=1 Absorption of photons with possible detection.

STRA This flag turns on the collision sampling method to simulate energy loss in thin materials, particularly gases. For more information see [PHYS334].

ISTRA =0 (**D**) Collision sampling switched off.
=1 Collision sampling activated.

SYNC Synchrotron radiation in magnetic field.

ISYNC =0 (**D**) The synchrotron radiation is not simulated.
=1 Synchrotron photons are generated, at the end of the tracking step.
=2 Photons are not generated, the energy is deposit locally.
=3 Synchrotron photons are generated, distributed along the curved path of the particle.

	Computation of total cross-section 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
Photoelectric effect	PHYS 230	PHYS 231
Photo fission of heavy elements	PHYS 240	PHYS 240
Rayleigh effect	PHYS 250	PHYS 251
Processes involving e^-/e^+		
Multiple scattering		PHYS 320 or 325 or 328
Ionisation and δ-rays production	PHYS 330	PHYS 331 or 332
Bremsstrahlung	PHYS 340	PHYS 341
Annihilation of positron	PHYS 350	PHYS 351
Generation of Čerenkov light	PHYS 260	PHYS 260
Synchrotron radiation	PHYS 360	
Processes involvingμ^-/μ^+		
Decay in flight	CONS 310	PHYS 400
Multiple scattering		PHYS 320 or 325
Ionisation and δ-rays production	PHYS 430	PHYS 331 or 332
Ionisation by heavy ions	PHYS 431	
Bremsstrahlung	PHYS 440	PHYS 441
Direct (e^+, e^-) pair production	PHYS 450	PHYS 451
Nuclear interaction	PHYS 460	PHYS 460
Generation of Čerenkov light	PHYS 260	PHYS 260
Processes involving hadrons		
Decay in flight	CONS 310	PHYS 400
Multiple scattering		PHYS 320 or 325
Ionisation and δ-rays production	PHYS 430	PHYS 331 or 332
Hadronic interactions	PHYS 500 or 510	PHYS 500 or 510
Generation of Čerenkov light	PHYS 260	PHYS 260

Table 1.1: Processes currently implemented in GEANT

Origin :
Revision :
Documentation : M. Maire, F.Carminati

Submitted: 20.12.84
Revised: 26.07.93

Compute the occurrence of a process

1 Principle

The simulation of the processes which accompany the propagation of a particle through the material of the detector (e.g. bremsstrahlung, δ-rays production, Compton scattering and so on) is performed by GEANT in the following steps:

1. Fetch a new particle to be tracked (often called *track* or *history*) from the stack supported by the link JSTAK (see [TRAK399]). This is done once at the beginning of each new track. The number of *interaction lengths* that the particle is going to travel, before undergoing each one of the possible discrete processes, is sampled at this point. These operations are done in the routine GLTRAC.
2. Evaluate the distance to the interaction point. This is done by the individual tracking routines (GTGAMA, GTNEUT, GTHADR, GTNINO, GTMUON, GTHION and GTCKOV) which control the tracking of particular particles. The number of interaction lengths remaining to travel before each of the possible processes (often called *tracking mechanisms* or simply *mechanisms*) is multiplied by the inverse of the macroscopic cross-section for that process in the current material (i.e. the interaction length). This gives the distances that the particle has to travel before each of the processes occurs in the current medium. The minimum among these numbers is the *step* over which the particle will be transported. In addition to the physics mechanisms, four *pseudo-interactions* are taken into account in the calculation of the step:
   (a) boundary crossing. The crossing of a volume boundary is treated like a discrete process. A particle never crosses a boundary during a step but rather stops there (NEXT mechanism);
   (b) maximum step limit. For each tracking medium a value for the maximum step can be specified by the user. Process SMAX;
   (c) maximum fraction of continuum energy loss, maximum angular deviation in magnetic field or maximum step for which the Molière formula, to simulate multiple scattering is valid. These are continuous processes, which introduce a limitation on the tracking step expressed by a single variable (see section [PHYS325] on GMULOF).
   (d) energy and time cut. Charged particles in matter are stopped when their energy falls below their energy threshold or when their time of flight exceeds the time cut;

   More information is given in the individual sections explaining the implementation of the physical processes.
3. Transport the particle either along a straight line (if no magnetic field or for a neutral particle) or along a helicoidal path (for charged particles in magnetic field).
4. Update the energy of the particle if continuous energy loss was in effect (charged particles in matter).
5. If a physical discrete process has been selected, generate the final state of the interaction.
6. If the incident particle *survives* the interaction (Compton, δ-rays production, bremsstrahlung, direct pair production by μ and μ-nucleus interaction, hadronic elastic scattering), sample again the number of interaction lengths to travel before the next event of the same kind. This is generally done by specialised routines: GMUNU, GCOMP, GBREM, etc.

7. Update the number of interaction lengths for all the processes and go back to (2) till the particle either leaves the detector or falls below its energy threshold or beyond its time cut or disappears in an interaction.

2 Distance evaluation

2.1 The interaction length

Let $\sigma(E, Z, A)$ be the total microscopic cross section for a given interaction. The mean free path, λ, for a particle to interact is given by:

$$\lambda = \frac{1}{\Sigma} \tag{1}$$

where Σ is the macroscopic cross-section in cm^{-1}. This quantity is given for an element by:

$$\Sigma = \frac{N_{Av}\rho\sigma(E, Z, A)}{A} \tag{2}$$

and for a compound or a mixture by:

$$\Sigma = \frac{N_{Av}\rho \sum_i n_i\sigma(E, Z_i, A_i)}{\sum_i n_i A_i} = N_{Av}\rho \sum_i \frac{p_i}{A_i}\,\sigma(E, Z_i, A_i) \tag{3}$$

N_{Av}	Avogadro's number (6.02486×10^{23})
Z	atomic number
A	atomic weight
ρ	density
σ	total cross-section for the reaction
n_i	proportion by number of the i^{th} element in the material
p_i	$= n_i A_i / \sum_j n_j A_j$, proportion by weight of the i^{th} element in the material

For electromagnetic processes which depend linearly on the atomic number Z we can write:

$$\begin{aligned}
\Sigma(E) &= N_{Av}\rho \sum_i \frac{p_i}{A_i}\,\sigma(E, Z_i) = N_{Av}\rho \sum_i \frac{p_i}{A_i}\,Z_i f(E) \\
&= N_{Av}\rho f(E) \sum_i \frac{p_i}{A_i}\,Z_i = N_{Av}\rho f(E) Z_{eff} \\
Z_{eff} &= \sum_i \frac{p_i}{A_i}\,Z_i
\end{aligned}$$

the value above of Z_{eff} is calculated by `GPROBI`. This mean free path is tabulated at initialisation time as a function of the kinetic energy of the particle, or, for hadronic interactions, it is calculated at tracking time.

Cross sections are tabulated in the energy range defined as: `ELOW(1)` $\leq E \leq$ `ELOW(NEK1)` in `NEK1` bins. These values can be redefined by the data record `RANGE`. Default values are `ELOW(1)` $= 10keV$, `ELOW(NEK1)` $= 10TeV$ and `NEKBIN` $=$ `NEK1` $- 1 = 90$. `NEKBIN` cannot be bigger than 199. The array `ELOW` is in the common `/GCMULO/`.

Numerically, if we measure the microscopic cross section in b where $1b = 10^{-24}cm^{-2}$, we can express the macroscopic cross section as:

$$\Sigma[cm^{-1}] = \frac{6.02486 \times 10^{23}\rho[g\ cm^{-3}]\sigma(E, Z, A)[b] \times 10^{-24}}{A} \tag{4}$$

$$= 0.602486\,\frac{\rho[g\ cm^{-3}]}{A}\,\sigma(E, Z, A)[b] \tag{5}$$

which is the formula mostly used in `GEANT`.

2.2 Determination of the interaction point

The mean free path of a particle for a given process, λ, depends on the medium and cannot be used directly to sample the probability of an interaction in a heterogeneous detector. The number of mean free paths which a particle travels is:

$$N_\lambda = \int \frac{dx}{\lambda(x)}$$

and it is independent of the material traversed. 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 function

$$P(N_R < N_\lambda) = 1 - e^{-N_\lambda}$$

The total number of mean free paths the particle travels before the interaction point, N_λ, is sampled at the beginning of the trajectory as:

$$N_\lambda = -\log(\eta)$$

where η is a random number uniformly distributed in the range $(0,1)$. N_λ is updated after each step Δx according the formula:

$$N'_\lambda = N_\lambda - \frac{\Delta x}{\lambda(x)}$$

until the step originating from $s(x) = N_\lambda \lambda(x)$ is the shortest and this triggers the specific process.

3 Common /GCPHYS/

The variables described above are stored in the common /GCPHYS/, one process per line:

```
      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
     +             ,IRAYL,SRAYL,SLRAYL,ZINTRA,STEPRA
      COMMON/GCPHLT/ILABS,SLABS,SLLABS,ZINTLA,STEPLA
     +             ,ISYNC
     +             ,ISTRA
*
```

The first 9 processes (from PAIR production up to MUon NUclear interaction) and the RAYLeigh scattering and Light ABSorbtion have the same scheme. Let's take as an example the first one (for a complete description of the common see [ZZZZ010]):

IPAIR flag for secondaries:

- 0= the process is turned off;
- 1= generation of secondaries enabled;

2= no generation of secondaries.

SPAIR $N_\lambda \lambda(x)$ = remaining track-length before interaction, evaluated at the last point where the mechanism was active, i.e. IPAIR$\neq 0$.

SLPAIR track length at the time when the interaction last happened for the current particle. Only SLPAIR (direct pair production by μ), SLDRAY (δ-ray production), SLBREM (bremsstrahlung for μ), SLHADR (hadronic interactions), SLMUNU (muon-nucleus interactions) and SLRAYL (Rayleigh effect) are used. The variables SOLOSS, STLOSS, SOMULS, STMULS are obsolete. They have been kept for backward compatibility, but their value is undefined and should not be used.

ZINTPA N_λ = remaining number of interaction lengths (mean free paths) evaluated at the last point where the mechanism was active, i.e. IPAIR$\neq 0$.

STEPPA $\lambda(x)$ = value of the interaction length at the last point where the mechanism was active, i.e. IPAIR$\neq 0$;

The evaluation and update of the quantities like STEPPA, SPAIR and ZINTPA are turned off in the media where the mechanism is not active (IPAIR $\neq 0$). Turning off a mechanism in one tracking medium may give incorrect physics results because not only will the mechanism not be active, but the interaction probabilities will not be updated, as if that medium had not been traversed at all. This feature of the tracking routines is used mainly in the vacuum, (defined as a medium with atomic number Z < 1), where all the mechanisms but DECAy (and SYNChrotron radiation, if activated) are inactive.

The DECAy in flight is simpler since the mean life time of the particle, τ, is not material dependent and can be sampled directly.

SLIFE not used.

SUMLIF proper time left before the decay. At the beginning of the track SUMLIF $= -c\tau \log(\eta)$.

SDCAY distance left to decay point evaluated at the last point where the mechanism was active, i.e. IDECA $\neq 0$.

4 Cross-section, energy loss and range tables

Cross-sections, energy loss dE/dx and range $R(E_{kin})$ are tabulated for all materials which enter in the definition of a tracking medium by the routine GPHYSI. The values of the energy for which the tabulated quantities are calculated are stored in the common /GCMULO/ (see [ZZZZ010]). To evaluate one of the tabulated quantities for a particle of *kinetic* energy E_0, a linear interpolation is used. Let i be such that:

$$E_i < E_0 \leq E_{i+1}$$

The integer variable IEKBIN in common /GCTRACK/ is equal to i during tracking and its value is recomputed by the routine GEKBIN when the energy of the particle changes. If the quantity Y has been tabulated so that $Y_i = Y(E_i)$ then the value $Y_0 = Y(E_0)$ is calculated as:

$$Y_0 = Y_i + \frac{E_0 - E_i}{E_{i+1} - E_i}(Y_{i+1} - Y_i) = Y_i \left(1 - \frac{E_0 - E_i}{E_{i+1} - E_i}\right) + Y_{i+1}\frac{E_0 - E_i}{E_{i+1} - E_i} \qquad (6)$$

Inside the code the following quantities are used:

$$\texttt{GEKRAT} = \frac{E_0 - E_i}{E_{i+1} - E_i} \qquad \texttt{GEKRT1} = \left(1 - \frac{E_0 - E_i}{E_{i+1} - E_i}\right)$$

where GEKRAT is in common /GCTRAK/ and GEKRT1 is a local variable recomputed when needed.

5 The energy loss tables

Energy loss and multiple scattering are continuous processes that are applied at every step for charged particles in matter ($Z \geq 1$).

As explained in [PHYS330] and [PHYS430], energy loss tables are calculated at initialisation time (GPHYSI) for all the materials which enter in the definition of a tracking medium (see [CONS200]). These tables contain NEK1 values of dE/dx calculated for the corresponding values of energy in ELOW.

In case of mixtures/compounds, the rule [7] is to combine the energy loss tables in GeV g^{-1} cm^2 according to the proportion by weight of the elements, that is:

$$\frac{dE}{dx} = \rho \sum_i \frac{p_i}{\rho_i} \left(\frac{dE}{dx} \right)_i$$

6 Limitations on the step size

The routine GMULOF called by GPHYSI creates and fills a table of NEK1 values corresponding to the ELOW values containing the smaller of the upper limits for the step imposed by the three continuous processes: energy loss, multiple scattering and bending of the track induced by the magnetic field.

Continuous energy loss can introduce an upper limit on the step via the variable DEEMAX, an argument to the GSTMED routine. During tracking the value of DEEMAX for the current medium is stored in the common /GCMATE/. DEEMAX is the maximum fraction of kinetic energy which a particle can lose in a step due to continuous ionisation ($0 < \mathtt{DEEMAX} < 1$). The limitation on the step size coming from DEEMAX is:

$$step \leq \frac{\mathtt{DEEMAX}}{dE/dx}$$

Multiple scattering as well can limit the step size, see [PHYS325]. The limitation is given as:

$$step \leq \min \left(T_{Bethe}, 10\, X_0 \right)$$

where X_0 is the radiation length and T_{Bethe} is the maximum step for which the Molière approximation is valid (see [PHYS325]).

Another upper limit on the step size comes from the magnetic field. The bending of the particle trajectory in the magnetic field may be limited by the TMAXFD argument to the GSTMED routine. During tracking the value of TMAXFD for the current medium is stored in the common /GCMATE/.

A lower limitation on the tracking step is not generally imposed. There is, however, a protection against the step being reduced to a very small value by continuous processes. In particular multiple scattering at low energies (< 1 MeV) can impose a very small tracking step with serious consequences on the tracking time. To avoid this, a lower limit on the step imposed by continuous processes is introduced: STMIN. The meaning of STMIN is the following: below 1 MeV the stopping range is usually small. If the stopping range becomes smaller than STMIN, the constraint imposed by the multiple scattering is ignored and the minimum is taken between the reduced stopping range (the distance the particle has to travel to reach its threshold energy) and STMIN itself. In this sense STMIN is no more than a tracking accelerator for stopping particles.

Another limitation on the step size which is imposed by the tracking routines during transport is the STEMAX parameter of GSTMED which sets an absolute upper limit to the size of a step for each tracking medium.

7 Automatic calculation of parameters

The definition of a *tracking medium* requires the specification by the user of a set of parameters (see [CONS200]) which can critically affect the tracking and hence the physics results of the GEANT MonteCarlo. To help the user to find the optimal set of parameters, by default GEANT overrides the values of STMIN, DEEMAX, STEMAX and TMAXFD. This behaviour is controlled by the AUTO data record and interactive command. By default AUTO=1 and automatic evaluation of the parameters is enforced (see below for the partly anomalous behaviour of TMAXFD). When AUTO= 0 then only those parameters which are < 0 are recalculated by GEANT, while for the others the user input is accepted with minimal checking.

When the automatic calculation of parameters is active, the following applies:

DEEMAX The formula used by GEANT is the following:

$$\mathtt{DEEMAX} = \begin{cases} 0.25 & \text{if } \mathtt{ISVOL} = 0 \text{ and } X_0 < 2\,\text{cm} \\ 0.25 - \dfrac{0.2}{\sqrt{X_0}} & \text{if } \mathtt{ISVOL} = 0 \text{ and } X_0 \geq 2\,\text{cm} \\ \dfrac{0.2}{\sqrt{X_0}} & \text{if } \mathtt{ISVOL} \neq 0 \end{cases}$$

ISVOL > 0 defines a *sensitive* detector, while ISVOL ≤ 0 is a non-*sensitive* detector (see [HITS]).

STMIN The default value corresponds to a stopping range of 200 keV above CUTELE.

TMAXFD The default value corresponds to 20° if the input value to GSTMED is ≤ 0 or $> 20^\circ$, otherwise the input value is taken.

STEMAX The default value is 10^{10} (BIG variable in common /GCONST/).

These values have been tuned empirically on a variety of setups and users are invited to start with automatic computation. The values of the parameters can be checked with a call to GPRINT('TMED',0) after GPHYSI has been called. Use of AUTO mode is strongly recommended because it makes GEANT a predictive tool. In this way all parameters are automatically computed by the system as opposed to tuning data and MonteCarlo via the tracking parameters.

8 Energy loss

8.1 Energy loss tables

In previous versions of GEANT (up to version 3.13) the energy lost by a charged particle travelling through matter was calculated by the formula:

$$\Delta E = \frac{dE}{dx} \times step$$

The quantity dE/dx was calculated and tabulated for $e^\pm$, $\mu^\pm$ and protons and the value for a particle of a given energy E_0 was calculated interpolating linearly the tables:

$$\begin{aligned} \left.\frac{dE}{dx}\right|_{E=E_0} &= \left.\frac{dE}{dx}\right|_{E=E_i} + \frac{E_0 - E_i}{E_{i+1} - E_i}\left(\left.\frac{dE}{dx}\right|_{E=E_{i+1}} - \left.\frac{dE}{dx}\right|_{E=E_i}\right) && (7) \\ &= \mathtt{GEKRT1} \times \left.\frac{dE}{dx}\right|_{E=E_i} + \mathtt{GEKRAT} \times \left.\frac{dE}{dx}\right|_{E=E_{i+1}} && (E_i < E_0 \leq E_{i+1}) \end{aligned}$$

These formula work of course on the assumption that the error is negligible. But the error on the linear approximation is $\mathcal{O}\left(\partial^2 E/\partial x\ \partial E\right)$ which is far from negligible at low energy. In this way the energy loss at low energies was constantly underestimated and the particles could travel too far before stopping.

8.2 Stopping range tables

To correct for this problem, a different approach was introduced in GEANT version 3.14. The *stopping range* of a particle is defined as the distance that the particle will travel before stopping. By definition the stopping range for a particle of energy E_0 is given by:

$$R = \int_{E_0}^{0} \frac{dx}{dE}\,dE = \int_{0}^{E_0} -\frac{dx}{dE}\,dE$$

Note that in the tables the positive quantity $-dE/dx$ is stored. The method used was to build a table of stopping ranges based on ELOW by integrating the inverse of the dE/dx tables in GRANGI. At tracking time the algorithm was the following:

1. Evaluate the stopping range for the threshold energy (STOPC). This was done only once at the beginning of each new track.

2. From the energy of the particle derive the stopping range by a linear interpolation of the range table:

   $$R_0 = \mathtt{GEKRT1} \times R_i + \mathtt{GEKRAT} \times R_{i+1}$$

   where $E_i < E_0 \leq E_{i+1}$.

3. Evaluate the stopping range for the particle after the step: $R'_0 = R_0 - step$. If this is less than the stopping range of a particle with threshold energy, the particle is terminated as a stopping particle below the energy cut. Otherwise the following quantities are evaluated:

   $$R_j < R'_0 \leq R_{j+1} \qquad \mathtt{GEK} = \frac{R'_0 - R_j}{R_{j+1} - R_j} \qquad \mathtt{GEK1} = 1 - \mathtt{GEK}$$

   and the final energy is computed as:

   $$E'_0 = \mathtt{GEK1} \times \mathtt{ELOW}(j) + \mathtt{GEK} \times \mathtt{ELOW}(j\mathtt{+1})$$

4. the energy loss is computed as:

   $$\Delta E = E_0 - E'_0$$

   This value is then corrected to take into account the energy loss fluctuations (see [PHYS332]).

This method has two main disadvantages. The first is due to the finite precision of computers. As the energy loss in a step is calculated as the difference of two numbers, it is subject to large relative errors. The effect can be particularly serious in the case of light materials, particles near the minimum ionisation point or with very short steps, where $\Delta E =$ DESTEP can even result in a negative quantity. As the relative precision of a 32-bit computer is around 10^{-6}, the error on the energy loss of a 100 GeV track can be as large as 100 keV.

The second problem connected with this method can be easily shown if we compute dE/dx as:

$$\begin{aligned}
\frac{dE}{dx} &= \frac{d}{dx}(E_0 - E'_0) = \frac{d}{dx}\left(E_0 - \mathtt{GEK1} \times \mathtt{ELOW}(j) - \mathtt{GEK} \times \mathtt{ELOW}(j\mathtt{+1})\right) \\
&= \frac{d}{dx}\left(E_0 - E_j - \frac{R'_0 - R_j}{R_{j+1} - R_j}(E_{j+1} - E_j)\right) = -\frac{d}{dx}\frac{\Delta E_j}{\Delta R_j}(R'_0 - R_j) \\
&= -\frac{\Delta E_j}{\Delta R_j}\frac{dR'_0}{dx} = \frac{\Delta E_j}{\Delta R_j}\frac{step}{dx} = \frac{\Delta E_j}{\Delta R_j}
\end{aligned}$$

As we can see, the reconstructed dE/dx curve due to continuous energy loss is a step function and constant in each energy bin. Thus, although the results obtained with GEANT 3.14 were very satisfactory, this was felt to be an undesirable feature.

8.3 Energy loss in GEANT

The two problems mentioned above have been solved in GEANT 3.15. As far as the precision is concerned, the solution was to revert to the algorithm of GEANT 3.13 every time the relative energy loss in the step (DESTEP/GEKIN) is smaller than five times the machine precision. This has given good results without loosing the substantial improvements obtained in GEANT 3.14 with the introduction of the stopping range algorithm. As a matter of fact, the above condition happens only the in the case of very small steps or when the dE/dx curve is very flat, and in both cases the algorithm based on the dE/dx tables is a good approximation.

The second problem has been solved by changing the interpolation algorithm, going from a linear to a quadratic interpolation. Instead of assuming a linear relation between energy and stopping range in every energy bin, we assume a quadratic relation of the sort:

$$E = f(R) = aR^2 + bR + c$$

The only problem now is the determination of the coefficients a, b and c. To do this we recall that the general formula of the parabola which passes through the points (x_1, y_1), (x_2, y_2), (x_3, y_3) is the following:

$$\begin{aligned}
y &= \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3 \\
&= x^2\left(\frac{y_1}{(x_1-x_2)(x_1-x_3)} + \frac{y_2}{(x_2-x_1)(x_2-x_3)} + \frac{y_3}{(x_3-x_1)(x_3-x_2)}\right) - \\
&\quad x\left(\frac{y_1(x_2+x_3)}{(x_1-x_2)(x_1-x_3)} + \frac{y_2(x_1+x_3)}{(x_2-x_1)(x_2-x_3)} + \frac{y_3(x_1+x_2)}{(x_3-x_1)(x_3-x_2)}\right) + \\
&\quad \left(\frac{y_1x_2x_3}{(x_1-x_2)(x_1-x_3)} + \frac{y_2x_1x_3}{(x_2-x_1)(x_2-x_3)} + \frac{y_3x_1x_2}{(x_3-x_1)(x_3-x_2)}\right)
\end{aligned}$$

This allows us to calculate and tabulate the coefficients a, b and c just by replacing y_i by `ELOW(I)` and x_i by the corresponding stopping range `ELOW(`i`)`. In the routine `GRANGI` a table of length $3 \times (\mathtt{NEKBIN} - 1)$ is created:

$$\begin{aligned}
\mathtt{Q(JINTRP+3*(I-1)+1)} &= a(R_I, R_{I+1}, R_{I+2}, E_I, E_{I+1}, E_{I+2}) = A_I \\
\mathtt{+2)} &= 0.5\,\frac{b(R_I, R_{I+1}, R_{I+2}, E_I, E_{I+1}, E_{I+2})}{a(R_I, R_{I+1}, R_{I+2}, E_I, E_{I+1}, E_{I+2})} = B_I \\
\mathtt{+3)} &= \frac{c(R_I, R_{I+1}, R_{I+2}, E_I, E_{I+1}, E_{I+2})}{a(R_I, R_{I+1}, R_{I+2}, E_I, E_{I+1}, E_{I+2})} = C_I
\end{aligned}$$

where E_I = `ELOW(I)` and R_I = `R(ELOW(I))`. The calculation of the energy loss is now done in the following steps:

1. Evaluate the stopping range for the threshold energy (`STOPC`), only once at the beginning of each new particle tracking.
2. From the energy of the particle derive the stopping range by a quadratic interpolation of the range table:

   $$R_0 = -B_I + \frac{A_I}{|A_I|}\sqrt{{B_I}^2 - \left(C_I - \frac{E_0}{A_I}\right)}$$

   The value of I is chosen according to the following. Let i be such that $\mathtt{ELOW}(i) < E_0 \leq \mathtt{ELOW}(i+1)$:

   $$\mathtt{I} = \begin{cases} \max(i-1, 1) & \text{if } \dfrac{E_0 - \mathtt{ELOW}(i)}{\mathtt{ELOW}(i+1) - \mathtt{ELOW}(i)} < 0.7 \\[2ex] \min(i, \mathtt{NEKBIN}-1) & \text{if } \dfrac{E_0 - \mathtt{ELOW}(i)}{\mathtt{ELOW}(i+1) - \mathtt{ELOW}(i)} \geq 0.7 \end{cases}$$

   The value of 0.7 is an empirical number which minimises the discontinuities of the reconstructed dE/dx curve.
3. Evaluate the stopping range for the particle after the step: $R_0' = R_0 - step$. If this is less than the stopping range of a particle of threshold energy, the particle is terminated as a stopping particle below the energy cut. Otherwise, the final energy is computed as:

   $$E_0' = A_I\left(C_I + R_0'\left(2B_I + R_0'\right)\right)$$

4. the energy loss is computed as:

$$\Delta E = E_0 - E_0'$$

This value is then corrected to take into account the energy loss fluctuations (see `[PHYS332]`).

This method results in a dE/dx curve which is a set of connected straight lines and not a step function.

Origin : R. Brun
Revision : F.Carminati
Documentation :

Submitted: 30.05.86
Revised: 15.12.93

Steering routine for physics initialisation

CALL **GPHYSI**

The routine GPHYSI is called at initialisation time by the user usually from the UGINIT routine (see example in [BASE100]). The purpose of GPHYSI is twofold:

- print the parameters of the current run
- compute the cross-section and energy loss tables and fill the JMATE data structure [CONS199].

1 Print the run definitions

GPHYSI prints the GEANT version number and the ZEBRA [8] version number stored in the JRUNG data structure [BASE299]

GPHYSI also prints the tracking and physics parameters for the current run, which are stored in the JTMED data structure. See the description of the JTMED data structure in [CONS210] and [CONS299].

A summary table like below will be output by GPHYSI

```
************************************************************
*                                                          *
*    G E A N T  Version 3.1600      DATE/TIME 930727/1135  *
*                                                          *
*                     R U N      1                        *
*                                                          *
************************************************************
*                                                          *
*      Data structure   Date   Time    GVERSN    ZVERSN    *
*      --------------   ----   ----    ------    ------    *
*                                                          *
*           INIT       930727  1135    3.1600     3.71     *
*                                                          *
*           KINE       930727  1135    3.1600     3.71     *
*                                                          *
*           HITS       930727  1135    3.1600     3.71     *
*                                                          *
*           DIGI       930727  1135    3.1600     3.71     *
*                                                          *
*     Random number seeds:          9876       54321       *
*     --------------------                                 *
*                                                          *
*----------------------------------------------------------*
*                                                          *
*              Standard TPAR for this run are              *
*              ------------------------------              *
*                                                          *
*  CUTGAM=100.00 keV  CUTELE=100.00 keV  CUTNEU=100.00 keV *
```

```
*  CUTHAD=100.00 keV  CUTMUO=  1.00 MeV                      *
*  BCUTE =100.00 keV  BCUTM =100.00 keV                      *
*  DCUTE =100.00 keV  DCUTM =100.00 keV  PPCUTM= 10.00 MeV *
*  IPAIR =         1.  ICOMP =         1.  IPHOT =         1. *
*  IPFIS =         1.  IDRAY =         1.  IANNI =         1. *
*  IBREM =         1.  IHADR =         1.  IMUNU =         1. *
*  IDCAY =         1.  ILOSS =         1.  IMULS =         1. *
*  IRAYL =         0.  ILABS =         0.  ISYNC =         0. *
*  ISTRA =         0.                                         *
*                                                             *
*                                                             *
*     Special TPAR for TMED   5   GAS                         *
*     -------------------------                               *
*  CUTGAM=100.00 keV  CUTELE=100.00 keV  CUTNEU=100.00 keV *
*  CUTHAD=100.00 keV  CUTMUO=  1.00 MeV                      *
*  BCUTE =100.00 keV  BCUTM =100.00 keV                      *
*  DCUTE = 10.00 GeV  DCUTM =100.00 keV  PPCUTM= 10.00 MeV *
*  IPAIR =         1.  ICOMP =         1.  IPHOT =         1. *
*  IPFIS =         1.  IDRAY =         1.  IANNI =         1. *
*  IBREM =         1.  IHADR =         1.  IMUNU =         1. *
*  IDCAY =         1.  ILOSS =         1.  IMULS =         1. *
*  IRAYL =         0.  ILABS =         0.  ISYNC =         0. *
*  ISTRA =         1.                                         *
*                                                             *
*                                                             *
***************************************************************
```

The meaning of the printed values is described in the documentation of the commons /GCCUTS/ and /GCPHYS/ in [ZZZZ010]. Different values of the parameters can be assigned by the user to each tracking medium (see [BASE100]). The parameter definition in GEANT proceeds as follows:

1. a default set of *global* values of the parameters is defined for all tracking media by the routine GINIT;
2. the default values can be changed by the user via the data records read by the routine GFFGO. A summary of all valid data records is given in [BASE040];
3. after having defined tracking media and materials, the user can redefine the parameters for **each medium** via the routine GSTPAR [CONS210];
4. if the data structures are read from an external file (see IOPA section), all the parameters are taken from this file. In this case the user can still modify them, but this should be done **before** calling the routine GPHYSI.

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 used as tracking media. GPHYSI builds and fills the JMATE data structure described in [CONS199]. Here, we give the flow chart of this process. The description of the specialised routines can be found in the rest of the section PHYS.

Note: if several tracking media are using the same material (for instance a calorimeter and a chamber support can be both in steel) the energy cuts must be the same. If this is not possible, the user must define materials with different numbers.

GPHYSI	Initialisation of physics processes. This routine should be called by the user before the tracking starts, but after all the material, *mechanism* flags and energy cuts have been defined. GPHYSI should also be called whenever a new initialisation data structure is read from disk.
GRNDMQ	Initialises the value of the seeds of the random number generator. See [BASE420] for more information
GEVKEV	Routine to format energy values for printout. See [BASE410] for more details.
GPHINI	Initialisation of the constants for photoelectric effect (see [PHYS230]).
	Loop on tracking media
GPHXSI	Initialisation of the cross section coefficients for the photoelectric effect in a tracking medium, see [PHYS230].
GPROBI	Initialisation of constants for various physical effects.
GMOLI	Initialises constants for Molière scattering, see [PHYS325].
	Loop on energy bins
GDRELA	Initialises the ionisation energy loss tables dE/dx for protons, μ and e^+, e^-, see [PHYS430].
GDRELP	Calculates ionisation energy loss for protons, see [PHYS430].
GDRELM	Calculates ionisation energy loss for μ, see [PHYS430].
GDRELE	Calculates ionisation energy loss for e^-/e^+, see [PHYS330].
GBRELA	Adds the contribution of bremsstrahlung to continuous energy loss tables for protons, μ and e^-/e^+, see [PHYS440].
GPRELA	Adds the contribution of direct pair production and μ-nucleus interactions to the μ continuous energy loss tables, see [PHYS450].
GPHOTI	Calculates the cross section for photoelectric effect, see [PHYS230].
GRAYLI	Initialises the tables of cross sections for the Rayleigh effect, see [PHYS250].
GANNII	Initialises the cross section for positron annihilation, see [PHYS350].
GCOMPI	Initialises the cross section tables for Compton effect, see [PHYS220].
GBRSGA	Initialises the cross section tables for discrete bremsstrahlung of e^-/e^+ and μ, see [PHYS340], [PHYS440].
GPRSGA	Initialises cross section tables for pair production by photons and direct pair production by muons, see [PHYS210], [PHYS450].
GDRSGA	Initialises the cross section tables for delta rays production by μ and electrons, see [PHYS330], [PHYS430].
GMUNUI	Initialises the cross section tables for μ-nucleus interactions, see [PHYS460].
GPFISI	Initialises the cross section tables for photo-fission and photo-absorption, see [PHYS240].
	End of loop on energy bins
GRANGI	Calculates the stopping range integrating the dE/dx tables, see [PHYS010].
GCOEFF	Calculation of the coefficients of the interpolating parabolas for the range tables, see [PHYS010].

GSTINI	Initialisation of the energy loss tables for the thin layers, see [PHYS334].
GFCNRM	Calculation of mean ionisation potentials and normalisation factors for oscillator functions.
GSREEO	Calculation of the real part of the refractive index.
GSDNDX	Calculation of number of collisions for a given value of β.
GMULOF	Calculation of the STMIN tables, see [PHYS325].
GMOLIO	Calculation of the constants of Molière scattering to estimate the step limitation due to multiple scattering simulation, see [PHYS325].

End of loop on tracking media

Notes:

1. The μ-nucleus interactions are treated **either** as a continuous energy loss by the muon (IMUNU$=0$) **or** as a discrete process (IMUNU≥ 1), **in an exclusive way** i.e.:
   - If IMUNU$=0$, the dE/dx due to the interactions is added to that coming from direct (e^+/e^-) pair production in the routine GPRELA.
   - 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 dependent on the nature of the incident particle. They are computed at tracking time by the function FLDIST in case of FLUKA [PHYS520], GHESIG in case of GHEISHA [PHYS510] or FMDIST in case of MICAP [PHYS530].

Origin : L. Urbán **Submitted:** 10.11.84
Revision : **Revised:** 16.12.93
Documentation :

Total cross-section for e^-e^+-pair production by photons

1 Subroutines

CALL **GPRSGA**

GPRSGA calculates the total cross-section for the pair production of photons and the direct pair production of muons in all materials. It tabulates the mean free path, $\lambda = 1/\Sigma$ (in cm), as a function of the medium and the energy. For the pair production by photons, it calls the function GPRSGG. The energy bins are the one of the array ELOW in common /CGMULO/ initialised in the routine GPHYSI. The following pointers are used:

JMA = LQ(JMATE-I)	pointer to the I^{th} material
JPAIR = LQ(JMA-10)	pointer to pair production cross-sections

GPRSGA is called at initialisation time by GPHYSI.

VALUE = **GPRSGG** (Z,E)

GPRSGG calculates the total cross-section for the pair production of a photon with energy E in material with atomic number Z. It is called by GPRSGA.

2 Method

We have parameterised the total cross-section of e^+e^--pair production by photon as:

$$\sigma(Z, E_\gamma) = Z(Z+1)\left(F_1(X) + F_2(X)Z + F_3(X)/Z\right) \quad \text{(barn/atom)} \tag{1}$$

where

$X = \ln(E_\gamma/m)$,
m = electron mass,
E_γ = photon energy

and

$$F_i(X) = \sum_{n=0}^{5} c_n X^n$$

The parameters c_n were taken from a least-square fit to the data [9] and can be found in the DATA statement in the function GPRSGG(Z,E) which computes the formula (1) (in barn/atom).
This parameterisation gives a good description of the data in the range:

$$\left.\begin{array}{r} 1 \leq Z \leq 100 \\ 1.5\ \text{MeV} \leq E_\gamma \leq 100\ \text{GeV} \end{array}\right\} \quad \frac{\Delta\,\sigma}{\sigma} \leq 5\% \qquad \text{with a mean value of} \approx 2.2\% \tag{2}$$

This mean free path is tabulated at initialisation as a function of the medium and of the energy by the routine GPRSGA.

Origin : G.N.Patrick, L.Urbán
Revision :
Documentation :

Submitted: 12.12.84
Revised: 16.12.93

Simulation of e^-e^+-pair production by photons

1 Subroutines

CALL **GPAIRG**

GPAIRG generates e^-e^+-pair production by photons. It uses a modified version of the random number techniques of Butcher and Messel [10] to sample the secondary electron/positron energies from the Coulomb corrected Bethe-Heitler differential cross-section. For the angular distribution of the pair, it calls the function GBTETH.

Input: via COMMON /GCTRAK/

Output: via COMMON /GCKING/

GPAIRG is called from GTGAMA, when the photon reaches its pair production point.

VALUE = **GBTETH** (ENER,PARTM,EFRAC)

GBTETH calculates the angular distribution of the e^-e^+-pair for pair production and of the photon for bremsstrahlung. In case of e^-e^+-pair production by photons, it gives the scaled angle for an electron (mass PARTM) of energy ENER which is EFRAC times the initial energy of the photon. GBTETH is called by GPAIRG.

2 Method

2.1 MonteCarlo technique

We give a very short summary of the random number technique used here [10, 11]. The method is a combination of the composition and rejection Monte Carlo methods. Suppose we wish to sample x from the distribution $f(x)$ and the (normalised) probability density function can be written as

$$f(x) = \sum_{i=1}^{n} \alpha_i f_i(x) g_i(x) \tag{1}$$

where $f_i(x)$ are normalised density functions, $\alpha_i > 0$ and $0 \leq g_i(x) \leq 1$.
According to this method, x can sampled in the following way:

1. select a random integer i such that $(1 \leq i \leq n)$ with probability proportional to α_i
2. select a value x' from the distribution $f_i(x)$
3. calculate $g_i(x')$ and accept $x = x'$ with probability $g_i(x')$;
4. if x' is rejected restart from step 1.

It can be shown that this scheme is correct 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 the 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 practical point of view (e.g. they can be very different in computational speed) and it can be very useful to optimise the decomposition. A remark of practical importance is that if our distribution is not normalised ($\int f(x)dx = C > 0; C \approx 1$), 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 $\sum_i \alpha_i / C$.

2.2 Differential cross-section for pair production

The Bethe-Heitler differential cross-section with the Coulomb correction for a photon of energy E to produce a $e^- e^+$-pair with one of the particles having energy ϵE (ϵ is the fraction of the photon energy carried by one particle of the pair) is given as in [12]:

$$\frac{d\sigma(Z,E,\epsilon)}{d\epsilon} = \frac{r_0^2 \alpha Z[Z+\xi(Z)]}{E^2} \left\{ [\epsilon^2 + (1-\epsilon)^2][\Phi_1(\delta) - \frac{F(Z)}{2}] + \frac{2}{3}\epsilon(1-\epsilon)[\Phi_2(\delta) - \frac{F(Z)}{2}] \right\} \quad (2)$$

where $\Phi_i(\delta)$ are the screening functions depending on the screening variable δ

$$\delta = \frac{136m}{Z^{1/3}E} \frac{1}{\epsilon(1-\epsilon)} \qquad m = \text{electron mass}$$

$$\left.\begin{array}{l} \Phi_1(\delta) = 20.867 - 3.242\delta + 0.625\delta^2 \\ \Phi_2(\delta) = 20.209 - 1.930\delta - 0.086\delta^2 \end{array}\right\} \quad \delta \leq 1$$

$$\Phi_1(\delta) = \Phi_2(\delta) = 21.12 - 4.184 \ln(\delta + 0.952) \quad \delta > 1$$

$$F(Z) = \begin{cases} 8/3 \ln Z & E < 0.05\ GeV \\ 8/3 \ln Z + 8 f_c(Z) & E \geq 0.05\ GeV \end{cases}$$

$$\xi(Z) = \frac{\ln(1440/Z^{2/3})}{\ln(183/Z^{1/3}) - f_c(Z)}$$

$f_c(Z)$ the Coulomb correction function

$$\begin{array}{lcl} f_c(Z) & = & a(1/(1+a) + 0.20206 - 0.0369a + 0.0083 * a^2 - 0.002a^3) \\ a & = & (\alpha * Z)^2 \\ \alpha & = & 1/137 \end{array}$$

The kinematical range for the variable ϵ is

$$\frac{m}{E} \leq \epsilon \leq 1 - \frac{m}{E} \quad (3)$$

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):

$$\frac{d\sigma}{d\epsilon} = \sum_{i=1}^{2} \alpha_i f_i(\epsilon) g_i(\epsilon)$$

where

$$
\begin{array}{llll}
\alpha_1 = \frac{(0.5-\epsilon_0)^2}{3}F_{10} & & \alpha_2 = \frac{1}{2}F_{20} & \\
f_1(\epsilon) = \frac{3}{(0.5-\epsilon_0)^3}(\epsilon-0.5)^2 & & f_2(\epsilon) = \frac{1}{0.5-\epsilon_0} & \\
g_1(\epsilon) = F_1/F_{10} & & g_2(\epsilon) = F_2/F_{20} &
\end{array}
$$

and

$$
\begin{array}{lll}
F_1 = F_1(\delta) = 3\Phi_1(\delta) - \Phi_2(\delta) - F(Z) & & F_{10} = F_1(\delta_{min}) \\
F_2 = F_2(\delta) = \frac{3}{2}\Phi_1(\delta) + \frac{1}{2}\Phi_2(\delta) - F(Z) & & F_{20} = F_2(\delta_{min})
\end{array}
$$

$$
\delta_{min} = 4\frac{136m}{Z^{\frac{1}{3}}3E}
$$

δ_{min} is the minimal value of the screening variable δ. It can be seen that the functions $f_i(\epsilon)$ are normalised and that the functions $g_i(\epsilon)$ are “valid" rejection functions.

Therefore, if r_i are uniformly distributed random numbers ($0 \leq r_i \leq 1$), we can sample the ϵ (x in the program) in the following way:

1. select i to be 1 or 2 according to the following ratio:

$$
\mathtt{BPAR} = \frac{\alpha_1}{\alpha_1 + \alpha_2}
$$

If $r_0 < \mathtt{BPAR}$ then $i = 1$, otherwise if $r_0 \geq \mathtt{BPAR}$ $i = 2$;

2. sample ϵ from $f_1(\epsilon)$. This can be performed by the following expressions:

$$
\epsilon = \begin{cases} 0.5 - (0.5-\epsilon_0)r_1^{\frac{1}{3}} & \text{when } i = 1 \\ \epsilon_0 + \left(\frac{1}{2} - \epsilon_0\right) r_1 & \text{when } i = 2 \end{cases}
$$

3. calculate the rejection function $g_i(\epsilon)$. If $r_2 \leq g_i(\epsilon)$, accept ϵ, otherwise return to step 1.

It should be mentioned that we need a step just after sampling ϵ in the step 2, because the cross-section formula becomes negative at large δ and this imposes an upper limit for δ;

$$
\delta_{max} = \exp\left[\frac{42.24 - F(Z)}{8.368}\right] - 0.952
$$

If we get a δ value using the sampled ϵ such that $\delta > \delta_{max}$, we have to start again from the step 1. After the successful sampling of ϵ, `GPAIRG` generates the polar angles of the electron with respect to an axis defined along the direction of the parent photon. The electron and the positron are assumed to have a symmetric angular distribution. The energy-angle distribution is given by Tsai [13, 14] as following:

$$
\begin{aligned}
\frac{d\sigma}{dpd\Omega} = & \frac{2\alpha^2 e^2}{\pi k m^4}\left\{\left[\frac{2x(1-x)}{(1+l)^2} - \frac{12lx(1-x)}{(1+l)^4}\right](Z^2+Z)\right. \\
& \left. + \left[\frac{2x^2-2x+1}{(1+l)^2} + \frac{4lx(1-x)}{(1+l)^4}\right][X - 2Z^2 f((\alpha Z)^2)]\right\}
\end{aligned}
$$

where k is the photon energy, p and E are the momentum and the energy of the electron of the e^+e^--pair, $x = E/k$ and $l = E^2\theta^2/m^2$. This distribution is quite complicated to sample and, furthermore, considered as function of the variable $u = E\theta/m$, it shows a very weak dependence on Z, E, k, $x = E/k$. Thus, the distribution can be approximated by a function

$$f(u) = C\left(ue^{-au} + due^{-3au}\right) \tag{4}$$

where

$$\begin{aligned} C &= \frac{9a^2}{9+d} \\ a &= 0.625 \\ d &= 27.0 \end{aligned}$$

The sampling of the function $f(u)$ can be done in the following way (r_i are uniformly distributed random numbers in [0,1]):

1. choose between ue^{-au} and due^{-3au}, with relative probability given by $9/(9+d)$ and $d/(9+d)$ respectively; if $r_1 < 9/(9+d)$ then $b = a$, else $b = 3a$;
2. sample ue^{-bu}, $u = -(\ln r_2 + \ln r_3)/b$.

The azimuthal angle, ϕ, is generated isotropically.

This information together with the momentum conservation is used to calculate the momentum vectors of both decay products and to transform them to the `GEANT` coordinate system. The choice of which particle in the pair is the electron/positron is made randomly.

2.3 Restrictions

1. Effects due to the breakdown of Born approximation at low energies are ignored (but the Coulomb correction is now included):
2. as suggested by Ford and Nelson [15], for very low energy photons ($E \le 2.1 MeV$) the electron energy is approximated by sampling from a uniform distribution over the interval $m \rightarrow 1/(2E)$. 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 practically 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 chemical 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 shown 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.

Origin : G.N.Patrick, L.Urbán
Revision :
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Total cross-section for Compton scattering

1 Subroutines

CALL **GCOMPI**

GCOMPI calculates the total cross-section for Compton scattering. It tabulates the mean free path, $\lambda = \frac{1}{\Sigma}$ (in cm), 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. GCOMPI is called at initialisation time by GPHYSI.

2 Method

The mean free path, λ, for a photon to interact via Compton scattering is given by

$$\lambda = \frac{1}{\Sigma} = \frac{A}{N_{Av}\rho} \frac{1}{\sigma(Z,E)}$$

where:

N_{Av}	Avogadro's number
Z, A	atomic and mass number of the medium
ρ	density of the medium
σ	total cross-section per atom for Compton scattering
E	energy of the photon.

For the total cross-section an empirical cross-section formula is used which reproduces the cross-section data rather well down to 10 keV:

$$\sigma(Z,E) = Z\left[P_1(Z)\frac{\log(1+2X)}{X} + \frac{P_2(Z) + P_3(Z)X + P_4(Z)X^2}{1 + aX + bX^2 + cX^3}\right] \text{barn atom}^{-1}$$

where

$$\begin{aligned} m &= \text{electron mass} \\ X &= \frac{E}{m} \\ P_i(Z) &= D_i + E_i Z + F_i Z^2 \end{aligned}$$

The values of the parameters are put in the DATA statement within the routine GCOMPI. The fit was made over 511 data points chosen between:

$$1 \leq Z \leq 100 \qquad ; \qquad 10\,\text{keV} \leq E \leq 100\,\text{GeV}$$

The accuracy of the fit is estimated to be:

$$\frac{\Delta\sigma}{\sigma} = \begin{cases} \approx 10\% & \text{for E} \simeq 10\,\text{keV} - 20\,\text{keV} \\ \leq 5-6\% & \text{for E} > 20\,\text{keV} \end{cases}$$

Origin : G.N.Patrick, L.Urbán
Revision :
Documentation :

Submitted: 26.09.83
Revised: 16.12.93

Simulation of Compton scattering

1 Subroutines

CALL **GCOMP**

GCOMP generates the Compton scattering of a photon on an atomic electron. It uses the random number techniques of Butcher and Messell [10] to sample the scattered photon energy according to the Klein-Nishina formula [16].

The interaction produces one electron, which is put in the /GCKING/ common block for further tracking. Tracking of the scattered photon will continue, with direction and energy changed by the interaction. All input/output information is through GEANT common blocks.

Input: via COMMON /GCTRACK/

Output: via COMMONs /GCTRAK/ and /GCKING/

Compton scattering is selected in GEANT by the input data record COMP. When Compton scattering is selected, GCOMP is called automatically from the GEANT photon tracking routine GTGAMA.

2 Method

For a complete account of the Monte Carlo methods used the interested user is referred to the publications of Butcher and Messel [10], Messel and Crawford [17] and Ford and Nelson [12]. Only the basic formalism is outlined here.

The quantum mechanical Klein-Nishina differential cross-section is:

$$\Phi(E,E') = \frac{X_0 n\pi r_0^2 m_e}{E^2}\left[\frac{1}{\epsilon}+\epsilon\right]\left[1-\frac{\epsilon\sin^2\theta}{1+\epsilon^2}\right]$$

where,
- E = energy of the incident photon
- E' = energy of the scattered photon
- ϵ = E'/E
- m_e = electron mass
- n = electron density
- r_0 = classical electron radius
- X_0 = radiation length

Assuming an elastic collision, the scattering angle θ is defined by the Compton formula:

$$E' = E\frac{m_e}{m_e + E(1-\cos\theta)}$$

Using the combined "composition and rejection" Monte Carlo methods described in chapter PHYS211, we may set:

$$
\begin{array}{lclll}
f(\epsilon) & = & \left[\frac{1}{\epsilon}+\epsilon\right] = \sum_{i=1}^{2}\alpha_i f_i(E) & \text{for} & \epsilon_0 > \epsilon > 1 \\
g(\epsilon) & = & \left[1-\frac{\epsilon\sin^2\theta}{1+\epsilon^2}\right] & \text{rejection function} & \\
\alpha_1 & = & \ln(1/\epsilon_0) & \alpha_2 & = \frac{1}{2}(1-\epsilon_0^2) \\
f_1(\epsilon) & = & \frac{1}{\epsilon\ln(1/\epsilon_0)} & f_2(\epsilon) & = \frac{2\epsilon}{1-{\epsilon_0}^2}
\end{array}
$$

The value of ϵ corresponding to the minimum photon energy (backward scattering) is given by:

$$\epsilon_0 = \frac{1}{1+2E/m_e}$$

Given a set of random numbers r_i uniformly distributed in [0,1], the sampling procedure for ϵ is the following:

1. decide which element of the $f(\epsilon)$ distribution to sample from. Let $\alpha_T = (\alpha_1+\alpha_2)r_0$. If $\alpha_1 \geq \alpha_T$ select $f_1(\epsilon)$, otherwise select $f_2(\epsilon)$;
2. sample ϵ from the distributions corresponding to f_1 or f_2. For f_1 this is simply achieved by:

   $$\epsilon = \epsilon_0 e^{\alpha_1 r_1}$$

   For f_2, we change variables and use:

   $$\epsilon' = \begin{cases} \max(r_2, r_3) & \text{for} \quad E/m \geq (E/m+1)r_4 \\ r_5 & \text{for all other cases} \end{cases}$$

   Then, $\epsilon = \epsilon_0 + (1-\epsilon_0)\epsilon'$;
3. calculate $\sin^2\theta = \max(0, t(2-t))$ where $t = m_e(1-\epsilon)/E'$
4. test the rejection function, if $r_6 \leq g(\epsilon)$ accept ϵ, otherwise return to step 1.

After the successful sampling of ϵ, GCOMP generates the polar angles of the scattered photon with respect to the direction of the parent photon. The azimuthal angle, ϕ, is generated isotropically and θ is as defined above. The momentum vector of the scattered photon is then calculated according to kinematic considerations. Both vectors are then transformed into the GEANT coordinate system.

3 Restriction

The differential cross-section is 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 [18], this has a negligible effect because of the small number of recoil electrons produced at very low energies.

Origin : L.Urbán
Revision : J. Chwastowski
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Total cross-section for photoelectric effect

1 Subroutines

CALL **GPHINI**

GPHINI steers initialisation of the constants. It calls GHPRIN for initialisation of the atom decay rates, GHSLIN to initialise atomic shell potentials and ionisation energies and GPHXIN for the initialisation of the cross-section constants. It is called by GPHOTI.

CALL **GPHRIN**

Initialisation of the atom decay rates. The routine is called by GPHINI.

CALL **GSHLIN**

Initialisation of the atomic shell potentials and ionisation energies. GSHLIN is called by GPHINI.

CALL **GPHXIN**

Initialisation of the cross-section constants for elements with $Z \leq 100$. It is called by GPHINI.

CALL **GPHXSI**

GPHXSI is called by GPHYSI for every tracking medium. It calculates the cross-section coefficients. The results of the calculations are stored in the photoelectric effect constants bank created by this routine.

CALL **GFSHDC**

GFSHDC calls GFRDT, GNRDT and GFSDPR for the probabilities of atomic state transitions and the transition energies and stores them in the ZEBRA bank. It is called by GPHXSI.

CALL **GFSHLS**

GFSHLS returns the shell potentials in eV for an atom. It is called by GFSHDC.

CALL **GFRDT**

GFRDT returns probability distribution and energies of the atomic radiative transitions. It is called by GFSHDC.

CALL **GNFRDT**

GNFRDT returns probability distribution and energies of the non-radiative atomic transitions. It is called by GFSHDC.

CALL **GFSDPR**

GFSDPR returns probability of the shell radiative decay. It is called by GFSHDC.

CALL **GPHOTI**

GPHOTI calculates the total cross-section for the photoelectric effect. It tabulates the mean free path, $\lambda = 1/\Sigma$ (in cm), as a function of the medium and the energy. If $Z \leq 100$ then it calls the function GPHSG1 for the total cross-section. Otherwise GPHSIG function is used. GPHOTI is called at initialisation time by GPHYSI.

VALUE = **GPHSG1** (EGAM)

GPHSG1 calculates the total cross-section for photoelectric effect. It is called by GPHOTI for $Z \leq 100$.

VALUE = **GPHSIG** (Z,EGAM)

GPHSIG calculates the total cross-section for the photoelectric effect of a photon with energy EGAM in material with atomic number $Z > 100$. It is called by GPHOTI.

2 Method

2.1 Materials with $Z \leq 100$

For media with $Z \leq 100$ we use SANDIA parametrisation [19]. The cross-section for elements was fitted with a linear combination of reciprocal powers of the photon energy E_γ (E_γ in keV). The fits were performed in different intervals *j* of the photon energy. In such an interval the cross-section reads as:

$$\mu_{ij} = \frac{C_{1,ij}}{E_\gamma} + \frac{C_{2,ij}}{E_\gamma^2} + \frac{C_{3,ij}}{E_\gamma^3} + \frac{C_{4,ij}}{E_\gamma^4} \quad (cm^2 g^{-1}) \tag{1}$$

with *j* changing from 1 to m_i, where m_i is the number of fitting intervals used for element *i*.

For the composites or mixtures of N elements the cross-section for j^{th} interval is calculated as:

$$\mu_j = \sum_{k=1}^{N} f_k \mu_{j,k} \tag{2}$$

where f_k is the fraction by mass of kth element in the material. Substituting (1) into (2) one finds that the cross-section coefficients can be calculated as:

$$C_{i,total} = \sum_{k=1}^{N} f_k C_{i,jk} \tag{3}$$

for `i` from 1 to 4. The macroscopic cross-section is calculated as follows:

$$\Sigma = \rho \mu \quad (cm^{-1}) \tag{4}$$

where ρ is the medium density.

As follows from (3) each compound is decomposed to its components which should coincide with chemical elements. If this is not the case i.e. a component has a non-integer atomic number Z_x and is created by a call to `GSMATE` then the cross-section constants are calculated using two elements with $Z_1 = integer(Z_x)$ and $Z_2 = Z_1 + 1$ applying the weights $w_1 = Z_2 - Z_x$ and $w_2 = Z_x - Z_1$ respectively.

2.2 Materials with $Z > 100$

Originally the parametrisation described below was developed for the elements with the atomic number `Z` between 5 and 100. Lacking more accurate data and assuming that there are no dramatic changes of the cross-section behaviour this method (i.e. `GPHSIG` function) is used for $Z > 100$.

The macroscopic cross-section for an element is given by

$$\Sigma = \frac{N \rho \sigma(Z, E_\gamma)}{A} \quad (cm^{-1}) \tag{5}$$

and for a compound or a mixture

$$\Sigma = \frac{N \rho \sum_i \sigma(Z_i, E_\gamma)}{\sum_i n_i A_i} = N \rho \sum_i \frac{p_i}{A_i} \sigma(Z_i, E_\gamma) \quad (cm^{-1}) \tag{6}$$

where

N	Avogadro's number
$Z(Z_i)$	atomic number (of ith component) of the medium
$A(A_i)$	atomic mass (of ith component) of the medium
ρ	density
σ	cross-section
n_i	proportion by number of the ith element in the material ($n_i = Wp_i/A_i$ where p_i is the corresponding proportion by weight and W is the molecular weight).

The binding energy of the inner shells has been parameterised as:

$$E_i(Z) = Z^2(A_i + B_iZ + C_iZ^2 + D_iZ^3) \quad \text{MeV} \tag{7}$$

where, $i = K, L_I, L_{II}$ and the constants A_i, B_i, C_i and D_i are tabulated below.

	A	B	C	D
K	$0.66644\cdot10^{-5}$	$0.22077\cdot10^{-6}$	$-0.32552\cdot10^{-8}$	$0.18199\cdot10^{-10}$
L_I	$-0.29179\cdot10^{-6}$	$0.87983\cdot10^{-7}$	$-0.12589\cdot10^{-8}$	$0.69602\cdot10^{-11}$
L_{II}	$-0.68606\cdot10^{-6}$	$0.10078\cdot10^{-6}$	$-0.14496\cdot10^{-8}$	$0.78809\cdot10^{-11}$

The photoelectric effect total cross-section has been parameterised as:

$$\sigma(Z,E) = Z^\alpha X^\beta \begin{cases} p_1/Z + p_2X + p_3 + p_4Z + p_5/X + p_6Z^2 & \\ \quad + p_7Z/X + p_8/X^2 + p_9Z^3 & \text{if} \quad E > E_k(Z) \\ \quad + p_{10}Z^2/X + p_{11}Z/X^2 + p_{12}/X^3 & \\ p_{13}/Z + p_{14}X + p_{15} & \text{if} \quad E_{L_I} < E \le E_k(Z) \\ p_{16}/Z + p_{17}X + p_{18} & \text{if} \quad E_{L_{II}} < E \le E_{L_I} \\ p_{19} & \text{if} \quad 10\ keV \le E \le E_{L_{II}} \end{cases} \tag{8}$$

where X is the ratio of the electron mass to the incident photon energy and σ is expressed in barns/atom. The fit was made over 301 data points chosen between $5 \le Z \le 100$ and $10\ keV \le E \le 50\ MeV$. The values of the parameters are put in a `DATA` statement within the function `GPHSIG` which computes the formula (8). The accuracy of the fit is estimated to be

$$\frac{\Delta}{\sigma} \le \begin{cases} 25\% & \text{near to the peaks} \\ 10\% & \text{elsewhere.} \end{cases}$$

3 Photoelectric effect Bank

The banks connected to the photoelectric effect are created during initialisation for each tracking medium. `PHXS` bank (data area) The total cross-section values are stored in `PHOT` bank in energy bins set within the array `ELOW` (common `CGMULO`). This bank is pointed by `JPHOT` link. The first structural link of `PHOT` supports the `PHXS` bank. For each energy interval of the cross-section we store the upper limit of the interval and four constants of equation (3). For each chemical element used to build the medium a `PHFN` bank is created. This bank, pointed by a structural link of the `PHXS` bank, contains the data needed for the photoelectric effect final state simulation. All data word are of the Fortran REAL type. The descriptions of the `PHXS` and `PHFN` banks are given below.

PHXS bank (data area)

Word #	Description
1	NZ-number of chemical elements of the medium
2 → NZ+1	atomic numbers of the elements
NZ+2 → 2*NZ+1	not used at present
2*NZ+2 → 3*NZ+1	weights of the cross-section constants
3*NZ+2	NIT-number of the cross-section intervals
3*NZ+3 → 1+3*NZ+1+5*NIT	the total cross-section constants

PHFN bank (data area)

Word #	Description
1	NIE-number of intervals for an element
2 → NIE*5+1	the element cross-section constants
NIE*5+2	number of shells used. At present always 4
NIE*5+3 → NIE*5+6	binding energies of K, L_I, L_{II} and L_{III} shells
NIE*5+7 → NIE*5+10	probability of the radiative shell decay
NIE*5+11	pointer to the radiative decays of K shell KRD
NIE*5+12	pointer to the radiative decays of L_I shell L1RD
NIE*5+13	pointer to the radiative decays of L_{II} shell L2RD
NIE*5+14	pointer to the radiative decays of L_{III} L3RD
NIE*5+15	pointer to the non-radiative decays of K shell KNRD
NIE*5+16	pointer to the non-radiative decays of L_I shell L1NRD
NIE*5+17	pointer to the non-radiative decays of L_{II} shell L2NRD
NIE*5+18	pointer to the non-radiative decays of L_{III} shell L3NRD
KRD	number of K shell radiative decay modes NRDK
KRD+1 → KRD+1+NRDK	K shell decay mode probability
KRD+1+NRDK → KRD+1+2*NRDK	K shell transition energies
L1RD	number of L_I shell radiative decay modes NRDL1
L1RD+1 → L1RD+1+NRDL1	L_I shell decay mode probability
L1RD+1+NRDL1 → L1RD+1+2*NRDL1	L_I shell transition energies
L2RD	number of L_I shell radiative decay modes NRDL2
L2RD+1 → L2RD+1+NRDL2	L_{II} shell decay mode probability
L2RD+1+NRDL2 → L2RD+1+2*NRDL2	L_{II} shell transition energies
L3RD	number of L_I shell radiative decay modes NRDL3
L3RD+1 → L3RD+1+NRDK	L_{III} shell decay mode probability
L3RD+1+NRDL3 → L3RD+1+2*NRDL3	L_{III} shell transition energies
KNRD	number of K shell radiative decay modes RDK = 1
KNRD+1 → KNRD+1+RDK	K shell decay mode probability
KNRD+1+RDK → KNRD+1+2*RDK	K shell transition energies
L1NRD	number of L_I shell radiative decay modes RDL1 = 1
L1NRD+1 → L1RD+1+RDL1	L_I shell decay mode probability
L1NRD+1+RDL1 → L1RD+1+2*RDL1	L_I shell transition energies
L2NRD	number of L_I shell radiative decay modes RDL2 = 1
L2NRD+1 → L2RD+1+NRDK	L_{II} shell decay mode probability

PHFN bank (data area, continued)

Word #	Description
L2NRD+1+RDL2 → L2RD+1+2*RDL2	L_{II} shell transition energies
L3NRD	number of L_I shell radiative decay modes RDL3 = 1
L3NRD+1 → L3RD+1+RDL3	L_{III} shell decay mode probability
L3NRD+1+RDL3 → L3RD+1+2*NRDL3	L_{III} shell transition energies

Origin : L.Urbán
Revision : J. Chwastowski
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Simulation of photoelectric Effect

1 Subroutines

```
CALL GPHOT
```

simulates the photoelectric effect. It uses the following input and output:

`Input` via common `GCTRAK`
`Output` via common `GCKING`

It calls the functions `GPHSG1`, `GPHSGP`, `GAVRL2` and `GAVRL3` for the cross-section and the functions `GHPHAK`, `GHPHAL1`, `GHPHAL2`, `GPHAL3` for the cosine distribution of the photoelectron. `GPHOT` is called by the tracking routine `GTGAMA`, if, and when, the photon reaches its interaction point.

```
VALUE = GAVRL2 (GAMAL2,BETAL2,X)
```

`GAVRL2` calculates approximation of the cross-section for the photoelectric effect from L_{II} shell. `GAMAL2`, `BETAL2` are the Lorentz gamma and beta factors of the emitted photoelectron and $X = m_e/E_\gamma$ where m_e is the electron mass and E_γ is the incident radiation energy. `GAVRL2` is called by `GHPOT`.

```
VALUE = GAVRL3 (GAMAL3,BETAL3,X)
```

`GAVRL3` calculates approximation of the cross-section for the photoelectric effect from L_{III} shell. `GAMAL3`, `BETAL3` are the Lorentz gamma and beta factors of the emitted photoelectron and $X = m_e/E_\gamma$ where m_e is the electron mass and E_γ is the incident radiation energy. `GAVRL3` is called by `GPHOT`.

```
VALUE = GPHAK (BETAK)
```

```
VALUE = GPHAL1 (BETAL1)
```

`GPHAK` and its entry point `GPHAL1` poll the cosine distribution (w.r.t. the incident photon direction) of the photoelectron emitted from K or L_I shell, respectively. They are called by `GPHOT`.

```
VALUE = GPHAL2 (BETAL2)
```

`GPHAL2` polls the cosine distribution (w.r.t. the incident photon direction) of the photoelectron emitted from L_{II} shell.

```
VALUE = GPHAL3 (BETAL3)
```

`GPHAL3` polls the cosine distribution (w.r.t. the incident photon direction) of the photoelectron emitted from L_{III} shell. It is called by `GPHOT`.

```
VALUE = GPHSGP (EGAM)
```

`GPHSGP` calculates the photoelectric effect total cross-section for a particular atom of a mixture taking into account weights. It is called by `GPHOT` for $Z \leq 100$.

2 Method

If the energy of the radiation incident on an atom is E_γ, then the quanta can be absorbed if $E_\gamma > E_{shell}$. The photoelectron is emitted with total energy:

$$E_{photoelectron} = E_\gamma - E_{shell} + m_e. \quad (1)$$

In the above equation it is assumed that the atom has infinite mass. One should note that there exists a correction term (see [20] and references therein) which becomes more and more important with increasing E_γ [21] [22] [23].

2.1 Probability of Interaction with an Atom

Probability of the interaction with an atom is calculated taking into account partial cross-sections of atoms of a mixture or a compound.

2.2 Probability of Interaction with a Shell

To calculate the probability of the interaction with a particular shell we use the jump ratios defined as:

$$J_{shell} = \frac{\sigma(E_{shell} + \delta E)}{\sigma(E_{shell} - \delta E)} \quad (2)$$

where $\delta E \to 0$. In addition we assume that the jump ratio is also valid away from the edges.
From (2) it follows that the probability p_{shell} to interact with a shell is:

$$p_{shell} = 1 - \frac{1}{J_{shell}} \quad (3)$$

We use the following parametrisation of the jump ratios for K and L_I shells [24]:

$$J_K = \frac{125}{Z} + 3.5 \quad (4)$$

$$J_{L_I} = 1.2 \quad (5)$$

For the L_{II} and L_{III} shells we adopt approximation of the formulae calculated by Gavrila [25]:

$$\sigma_{L_{II}} = \gamma\beta\frac{m_e}{E_\gamma}\left\{\gamma^3 - 5\gamma^2 + 24\gamma - 16 - (\gamma^2 + 3\gamma - 8)\frac{\log(\gamma(1+\beta))}{\gamma\beta}\right\} \quad (6)$$

and

$$\sigma_{L_{III}} = \gamma\beta\frac{m_e}{E_\gamma}\left\{4\gamma^3 - 6\gamma^2 + 5\gamma + 3 - (\gamma^2 - 3\gamma + 4)\frac{\log(\gamma(1+\beta))}{\gamma\beta}\right\} \quad (7)$$

where

γ, β	are the emitted photoelectron Lorentz gamma and beta factors;
E_γ	is the incident radiation energy;
m_e	is the electron mass.

Below an example of the shell interaction probability calculations for $E_\gamma > E_K$ is given.
If

$$\begin{aligned}
\Sigma_{II,III} &= \sigma_{L_{II}} + \sigma_{L_{III}} \\
r_{L_{II}} &= \frac{\sigma_{L_{II}}}{\Sigma_{II,III}} \\
r_{L_{III}} &= \frac{\sigma_{L_{III}}}{\Sigma_{II,III}}
\end{aligned}$$

then from (3) one can find that

$$
\begin{aligned}
p_K &= 1 - \frac{1}{J_K} \\
p_{L_1} &= (1 - p_K) \cdot (1 - \frac{1}{J_{L_1}}) \\
p_{L_{II}} &= (1 - p_K - p_{L_I}) \cdot r_{L_{II}} \\
p_{L_{III}} &= (1 - p_K - p_{L_I}) \cdot r_{L_{III}}
\end{aligned}
$$

After simple calculations one obtains the probability distribution function which is used to select the shell.

2.3 Angular distributions of photoelectrons

The angular distributions of photoelectrons should be calculated using the partial wave analysis for the outgoing electron. Since this method is very time consuming we use approximations of the angular distributions calculated by F. von Sauter [26] [27] (K shell) and Gavrila [28] [25] (K and L shells). We use the cross-section which is correct only to zero order in αZ. This approximation foresees zero scattering amplitude in the direction of incident photon while experimentally the cross-section at 0 angle is non-vanishing [29]. If

$$
X = 1 - \beta \cos \Theta
$$

then for K and L_I shells we use:

$$
\frac{d\sigma_{K,L_I}}{d(\cos\Theta)} \sim \frac{\sin^2\Theta}{X^4}\left\{1 + \frac{1}{2}\gamma(\gamma-1)(\gamma-2)\right\} \tag{8}
$$

and for L_{II} and L_{III} shells we have:

$$
\begin{aligned}
\frac{d\sigma_{L_{II}}}{d(\cos\Theta)} \sim\ & \frac{(\gamma^2-1)^{\frac{1}{2}}}{\gamma^5(\gamma-1)^5}\left\{\frac{\gamma(3\gamma+1)}{2X^4} - \frac{\gamma^2(9\gamma^2+30\gamma-7)}{8X^3}\right. \\
& + \frac{\gamma^3(\gamma^3+6\gamma^2+11\gamma-2)}{4X^2} - \frac{\gamma^4(\gamma-1)(\gamma+7)}{8X} \\
& \left. + \sin^2\Theta\left(\frac{(\gamma+1)}{X^5} - \frac{\gamma(\gamma+1)}{X^4} - \frac{\gamma^2(3\gamma+1)(\gamma^2-1)}{X^3}\right)\right\}
\end{aligned} \tag{9}
$$

$$
\begin{aligned}
\frac{d\sigma_{L_{III}}}{d(\cos\Theta)} \sim\ & \frac{(\gamma^2-1)^{\frac{1}{2}}}{\gamma^5(\gamma-1)^5}\left\{-\frac{\gamma(3\gamma-1)}{2X^4} + \frac{\gamma^2(3\gamma^2-1)}{X^3}\right. \\
& \frac{\gamma^3(\gamma^3-3\gamma^2+2\gamma+1)}{X^3} - \frac{\gamma^4(\gamma-2)(\gamma-1)}{2X} \\
& \left. + \sin^2\Theta\left(\frac{(\gamma+1)}{X^5} - \frac{\gamma(\gamma+1)(3\gamma-1)}{X^4} + \frac{\gamma^2(\gamma^2-1)}{X^3}\right)\right\}
\end{aligned} \tag{10}
$$

The azimuthal angle distribution is uniform.

2.4 Final State

After the photoelectron emission the atom is left in excited state. The excitation energy equal to the binding energy E_i of the shell in which the interaction took place. Subsequently the atom emits a fluorescent photon or Auger or Coster-Kronig electron. The selection of radiative or non-radiative transition is based on compilation by Krause [30].

The Auger or Coster-Kronig transitions are represented by the most probable line for a given vacancy [31]. The emitted electron energy E_e is

$$
E_e = E_i - (E_j + E_k) \tag{11}
$$

where E_i, E_j, E_k are the subshell binding energies and $E_j > E_k$.
In case of fluorescence we use transition rates of Scofield [32]. We use only those transitions for which the occurrence probability is not less than 1%. The fluorescent photon is emitted with energy E_γ

$$E_\gamma = E_i - E_j \tag{12}$$

for transition between the subshells i and j.
In addition to the above, to fulfill the energy conservation law, emission of an additional photon is simulated. For non-radiative transitions its energy is E_k (see formula 11). In case of fluorescent transition this photon has energy E_j (see equation 12). The angular distribution of the emitted particle is isotropic.

Origin : H.C.Fesefeldt, F.Carminati
Revision :
Documentation :

Submitted: 20.12.85
Revised: 16.12.93

Photon-induced fission on heavy materials

1 Subroutines

```
CALL GPFISI
```

GPFISI computes and stores in the appropriate bank (see JMATE data structure) the value of the photo-fission cross sections as a function of the energy for all the material for which this process is activated and with $230 \leq A \leq 240$. It is called during initialisation by GPHYSI.

```
CALL GPFIS
```

GPFIS is called by the tracking routine GTGAMA to generate the final state of a photon-induced fission.

2 Method

This set of routines generates the photon-induced fission in the nuclei of heavy materials. The code is a straight translation into the GEANT style of the corresponding code from the GHEISHA Monte Carlo program. The information contained in this chapter is mainly taken from the GHEISHA manual to which the user is referred [1]. 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 between 230 and 240. The fission cross-section for uranium is tabulated, and for other elements the parameterisation of the A-dependence at 3 MeV neutron energy is used:

$$\sigma_f(3MeV) = -67.0 + 38.7\, Z^{4/3} A$$

which implies that the fission may occur only for $Z \geq 90$. The fission products are generated in an approximate way. The giant resonance effect is included here as nuclear photoabsorbtion. The photo-fission is activated by the data record PFIS. Secondaries coming from the interaction of the virtual photon with the nucleus will be in the GEANT temporary stack. If IPFIS=2 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.

Origin : G.Tromba, P.Bregant
Revision :
Documentation :
Submitted: 10.10.89
Revised: 16.12.93

Total cross-section for Rayleigh scattering

1 Subroutines

CALL **GRAYLI**

`GRAYLI` computes the mean free path, λ, for Rayleigh scattering as a function of the medium and of the energy. It evaluates also an integral of atomic form factors which is used in the routine `GRAYL` (see `[PHYS251]`) to sample the scattering angles. `GRAYLI` is called during initialization by `GPHYSI`.

2 Method

2.1 Total cross-section

An empirical cross-section is used to produce the total cross-section data [12, 33]:

$$\sigma_c(Z,E) = aE^3 + bE^2 + cE + d \qquad \text{barn/atom} \tag{1}$$

The values of the coefficients are stored in the `DATA` statement within the routine `GRAYLI`. For each element the fit was obtained over 27 experimental values of the total coherent cross-section. The accuracy of the fit is estimated to be about $\Delta\sigma/\sigma \approx 10\%$, but it is better for most of the elements.

2.2 The atomic form factors

Different empirical formulae are used to produce the atomic form factor F(Z,E) data [34]. For $Z = 1-3, 91-99$ the empirical expression is:

$$F(Z,E) = a_1E^7 + b_1E^6 + c_1E^5 + d_1E^4 + e_1E^3 + f_1E^2 + g_1E + h_1 \tag{2}$$

for Z=100 the formula used is:

$$F(Z,E) = a_2E^5 + b_2E^4 + c_2E^3 + d_2E^2 + e_2E + f_2 \tag{3}$$

for the other elements F is calculated as:

$$F(Z,E) = \begin{cases} a_3E^3 + b_3E^2 + c_3E + d_3 & \text{if } E \le E_c \\ a_4E^3 + b_4E^2 + c_4E + d_4 & \text{if } E > E_c \end{cases} \tag{4}$$

The value of the energy E_c depends on Z and it is set in the array `ELIM`. The values of the coefficients are stored in the `DATA` statement in the routine `GRAYLI`. For each element the fits were performed over 97 tabulated values of the form factors. The accuracy of the fit is estimated to about $\Delta\sigma/\sigma = 10\%$ for $E \le 1$ MeV for most of the elements. The integral over the atomic form factors $A(E_i)$ is defined as follows:

$$A(E_i) = 2\int_0^{E_i} |F_T(E)|^2 E\,dE = \int_0^{E_i} |F_T(E)|^2 dE^2 \tag{5}$$

where $F_T(E)$ is the total atomic form factor which is function of the energy E.

Origin : G.Tromba, P.Bregant
Revision :
Documentation :

Submitted: 10.10.89
Revised: 16.12.93

Simulation of Rayleigh scattering

```
CALL GRAYL
```

GRAYL generates Rayleigh scattering of a photon using the random-number composition and rejection technique to sample the momentum of the scattered photon and the scattering angle, according to the form-factor distribution. In this reaction no new particles are generated and the kinematical quantities of the scattered photon replace the original ones in the /GCTRAK/ common block.

Activation of the Rayleigh scattering is done via the FFREAD data record RAYL. If this process is activated, GRAYL is called by GTGAMA when a Rayleigh scattering occurs.

1 Method

The Rayleigh differential cross section as a function of q^2 is given by [35]:

$$\frac{d\sigma_R\left(q^2\right)}{d\Omega} = \frac{\pi r_0^2}{k^2}\left(\frac{1+\mu^2}{2}\right)\left|F_T\left(q\right)\right|^2 \tag{1}$$

where:

r_0	electron radius
k	incident wave vector
$q = 2ksin\frac{\theta}{2}$	momentum of scattered photon (θ is the scattering angle)
$\mu = cos\theta$	$= 1 - \frac{q^2}{2k^2}$
$F_T\left(q\right)$	molecular form factor

Under the assumption that the atoms of a molecule are completely independent, $\left|F_T\left(q\right)\right|^2$ is given by:

$$\left|F_T\left(q\right)\right|^2 = \sum_{i=1}^{N}\frac{W_i}{A_i}\left|F_i\left(q_i, Z_i\right)\right|^2\sigma_{c_i}\left(Z_i, E\right) \tag{2}$$

where the index i runs on the N atoms in the molecule and:

W_i	proportion by weight
Z_i, A_i	atomic number and weight
F_i	form factor
σ_{c_i}	total atomic cross section for coherent scattering

Using the combined composition and rejection sampling method described in GPAIRG ([PHYS211]) we may set:

$$f\left(q\right) = \sum_{i=1}^{N}\alpha_i f_i\left(q\right)g_i\left(q\right) = \sum_{i=1}^{N}A\left(q_i^2\right)\frac{\left|F_T\left(q\right)\right|^2}{A\left(q_n^2\right)}\left(\frac{1+\mu^2}{2}\right) \tag{3}$$

where:

n	number of energy bins
q_i	momentum of the photon with energy E_i of the i^{th} bin
q_n	upper limit for the momentum of the scattered photon
α_i	$A\left(q_i^2\right)$
$f_i\left(q\right)$	$\dfrac{\left\|F_T\left(q\right)\right\|^2}{A\left(q_n^2\right)}$
$g_i\left(q\right)$	$= \dfrac{1+\mu^2}{2}$ rejection function.

Therefore, for given values of the random numbers r_1 and r_2, `GRAYL` samples the momentum of the scattered photon and the scattering angle θ via the following steps:

1. sample $A\left(q^2\right) = r_1 A\left(q_n^2\right)$
2. find the $\left(q_{i-1}, q_i\right]$ interval which gives $A\left(q_{i-1}^2\right) \leq A\left(q^2\right) \leq A\left(q_i^2\right)$
3. calculate the linear extrapolation:

$$q = q_{i-1} + \left(A\left(q^2\right) - A\left(q_{i-1}^2\right)\right) \frac{q_i - q_{i-1}}{A\left(q_i^2\right) - A\left(q_{i-1}^2\right)} \tag{4}$$

4. calculate $\mu = cos\theta = 1 - q^2/(2k^2)$
5. calculate $g_i\left(q\right) = (1+\mu^2)/2$
6. if $g_i\left(q\right) > r_2$ the event is accepted, otherwise go back to 1.

Origin : F.Carminati, R.Jones
Revision :
Documentation : F. Carminati

Submitted: 03.02.93
Revised: 16.12.93

Čerenkov photons

Čerenkov photons are produced when a charged particle traverses a dielectric material.

1 Physics processes for optical photons

A photon is called optical when its wavelength is much greater than the typical atomic spacing, for instance when $\lambda \geq 10nm$ which corresponds to an energy $E \leq 100eV$. Production of an optical photon in a HEP detector is primarily due to:

1. Čerenkov effect;
2. Scintillation;
3. Fluorescence.

Fluorescence is taken into account in `GEANT` in the context of the photoelectric effect (`[PHYS230]`, `[PHYS231]`), but only above the energy cut `CUTGAM`. Scintillation is not yet simulated by `GEANT`.

Optical photons undergo three kinds of interactions:

1. Elastic (Rayleigh) scattering;
2. Absorption;
3. Medium boundary interactions.

1.1 Elastic scattering

For optical photons elastic scattering is usually unimportant. For $\lambda = .200\mu m$ we have $\sigma_{Rayleigh} \approx .2b$ for N_2 or O_2 which gives a mean free path of ≈ 1.7 km in air and ≈ 1 m in quartz. An important exception to this is found in aerogel, which is used as a Čerenkov radiator for some special applications. Because of the spectral properties of this material, Rayleigh scattering is extremely strong and this limits its usefulness as a RICH radiator. At present, elastic scattering of optical photons is not simulated in `GEANT`.

1.2 Absorption

Absorption is important for optical photons because it determines the lower λ limit in the *window* of transparency of the radiator. Absorption competes with photo-ionisation in producing the signal in the detector, so it must be treated properly in the tracking of optical photons.

1.3 Medium boundary effects

When a photon arrives at the boundary of a dielectric medium, its behaviour depends on the nature of the two materials which join at that boundary:

- Case dielectric → dielectric.
  The photon can be transmitted (refracted ray) or reflected (reflected ray). In case where the photon can only be reflected, total internal reflection takes place.

- Case dielectric $\rightarrow$ metal.
  The photon can be absorbed by the metal or reflected back into the dielectric. If the photon is absorbed it can be detected according to the photoelectron efficiency of the metal.
- Case dielectric $\rightarrow$ black material.
  A *black* material is a tracking medium for which the user has not defined any optical property. In this case the photon is immediately absorbed undetected.

2 Photon polarisation

The photon polarisation is defined as a two component vector normal to the direction of the photon:

$$\begin{pmatrix} a_1 e^{i\phi_1} \\ a_2 e^{i\phi_2} \end{pmatrix} = e^{\phi_o} \begin{pmatrix} a_1 e^{i\phi_c} \\ a_2 e^{-i\phi_c} \end{pmatrix}$$

where $\phi_c = (\phi_1 - \phi_2)/2$ is called circularity and $\phi_o = (\phi_1 + \phi_2)/2$ is called overall phase. Circularity gives the left- or right-polarisation characteristic of the photon. RICH materials usually do not distinguish between the two polarisations and photons produced by the Čerenkov effect are linearly polarised, that is $\phi_c = 0$. The circularity of the photon is ignored by `GEANT`.

The overall phase is important in determining interference effects between coherent waves. These are important only in layers of thickness comparable with the wavelength, such as interference filters on mirrors. The effects of such coatings can be accounted for by the empirical reflectivity factor for the surface, and do not require a microscopic simulation. `GEANT` does not keep track of the overall phase.

Vector polarisation is described by the polarisation angle $\tan\psi = a_2/a_1$. Reflection/transmission probabilities are sensitive to the state of linear polarisation, so this has to be taken into account. One parameter is sufficient to describe vector polarisation, but to avoid too many trigonometrical transformations, a unit vector perpendicular to the direction of the photon is used in `GEANT`.

The polarisation vectors are stored in a special track structure which is lifted at the link `LQ(JSTACK-1)` when the first Čerenkov photon is stored in the stack.

3 Method

3.1 Generation of the photons

For the formulas contained in this chapter, please see [36]. Let n be the refractive index of the dielectric material acting as a radiator ($n = c/c'$ where c' is the group velocity of light in the material: it follows that $1 \leq n$). In a dispersive material n is an increasing function of the photon momentum p_γ, $dn/dp \geq 0$. A particle travelling with speed $\beta = v/c$ will emit photons at an angle θ with respect to its direction, where θ is given by the relation:

$$\cos\theta = \frac{1}{\beta n}$$

from which follows the limitation for the momentum of the emitted photons:

$$n(p_\gamma^{min}) = \frac{1}{\beta}$$

Additionally, the photons must be within the window of transparency of the radiator. All the photons will be contained in a cone of opening $\cos\theta_{max} = 1/(\beta n(p_\gamma^{max}))$.

The average number of photons produced is given by the relations (Gaussian units):

$$dN = \frac{2\pi z^2 e^2}{\hbar c}\sin^2\theta \frac{d\nu}{c}dx = \frac{2\pi z^2 e^2}{\hbar c}\left(1 - \cos^2\theta\right)\frac{d\nu}{c}dx = \frac{2\pi z^2 e^2}{\hbar c}\left(1 - \frac{1}{n^2\beta^2}\right)\frac{d\nu}{c}dx =$$

$$= \frac{z^2 e^2}{\hbar^2 c^2}\left(1 - \frac{1}{n^2\beta^2}\right) dp_\gamma\, dx \approx 370 z^2 \frac{photons}{cm\ eV}\left(1 - \frac{1}{n^2\beta^2}\right) dp_\gamma\, dx$$

and

$$\frac{dN}{dx} \approx 370 z^2 \int_{p_\gamma^{min}}^{p_\gamma^{max}} dp_\gamma \left(1 - \frac{1}{n^2\beta^2}\right) = 370 z^2 \left(p_\gamma^{max} - p_\gamma^{min} - \frac{1}{\beta^2}\int_{p_\gamma^{min}}^{p_\gamma^{max}} dp_\gamma \frac{1}{n(p_\gamma)^2}\right)$$

The number of photons produced is calculated from a Poissonian distribution with average value $\bar{n}$ = STEP dN/dx. The momentum distribution of the photon is then sampled from the density function:

$$f(p_\gamma) = \left(1 - \frac{1}{n^2(p_\gamma)\beta^2}\right)$$

3.2 Tracking of the photons

Čerenkov photons are tracked in the routine GTCKOV. These particles are subject to *in flight* absorption (process LABS, number 101) and *boundary action* (process LREF, number 102, see above). As explained above, the status of the photon is defined by 2 vectors, the photon momentum ($\vec{p} = \hbar\vec{k}$) and photon polarisation ($\vec{e}$). By convention the direction of the polarisation vector is that of the electric field. Let also $\vec{u}$ be the normal to the material boundary at the point of intersection, pointing out of the material which the photon is leaving and toward the one which the photon is entering. The behaviour of a photon at the surface boundary is determined by three quantities:

1. refraction or reflection angle, this represents the kinematics of the effect;
2. amplitude of the reflected and refracted waves, this is the dynamics of the effect;
3. probability of the photon to be refracted or reflected, this is the quantum mechanical effect which we have to take into account if we want to describe the photon as a particle and not as a wave;

As said above, we distinguish three kinds of boundary action, dielectric $\rightarrow$ black material, dielectric $\rightarrow$ metal, dielectric $\rightarrow$ dielectric. The first case is trivial, in the sense that the photon is immediately absorbed and it goes undetected.

To determine the behaviour of the photon at the boundary, we will at first treat it as an homogeneous monochromatic plane wave:

$$\begin{aligned}
\vec{E} &= \vec{E}_0 e^{i\vec{k}\cdot\vec{x} - i\omega t} \\
\vec{B} &= \sqrt{\mu\epsilon}\frac{\vec{k}\times\vec{E}}{k}
\end{aligned}$$

Case dielectric $\rightarrow$ dielectric

In the classical description the incoming wave splits into a reflected wave (quantities with a double prime) and a refracted wave (quantities with a single prime). Our problem is solved if we find the following quantities:

$$\begin{aligned}
\vec{E'} &= \vec{E'}_0 e^{i\vec{k'}\cdot\vec{x} - i\omega t} \\
\vec{E''} &= \vec{E''}_0 e^{i\vec{k''}\cdot\vec{x} - i\omega t}
\end{aligned}$$

For the wave numbers the following relations hold:

$$
\begin{aligned}
|\vec{k}| &= |\vec{k}''| = k = \frac{\omega}{c}\sqrt{\mu\epsilon} \\
|\vec{k}'| &= k' = \frac{\omega}{c}\sqrt{\mu'\epsilon'}
\end{aligned}
$$

Where the speed of the wave in the medium is $v = c/\sqrt{\mu\epsilon}$ and the quantity $n = c/v = \sqrt{\mu\epsilon}$ is called *refractive index* of the medium. The condition that the three waves, refracted, reflected and incident have the same phase at the surface of the medium, gives us the well known Fresnel law:

$$
\begin{aligned}
(\vec{k}\cdot\vec{x})_{surf} &= (\vec{k}'\cdot\vec{x})_{surf} = (\vec{k}''\cdot\vec{x})_{surf} \\
k \sin i &= k' \sin r = k'' \sin r'
\end{aligned}
$$

where i, r, r' are, respectively, the angle of the incident, refracted and reflected ray with the normal to the surface. From this formula the well known condition emerges:

$$
\begin{aligned}
i &= r' \\
\frac{\sin i}{\sin r} &= \sqrt{\frac{\mu'\epsilon'}{\mu\epsilon}} = \frac{n'}{n}
\end{aligned}
$$

The dynamic properties of the wave at the boundary are derived from Maxwell's equations which impose the continuity of the normal components of $\vec{D}$ and $\vec{B}$ and of the tangential components of $\vec{E}$ and $\vec{H}$ at the surface boundary. The resulting ratios between the amplitudes of the the generated waves with respect to the incoming one are expressed in the two following cases:

1. a plane wave with the electric field (polarisation vector) perpendicular to the plane defined by the photon direction and the normal to the boundary:

$$
\begin{aligned}
\frac{E_0'}{E_0} &= \frac{2n\cos i}{n\cos i + \frac{\mu}{\mu'}n'\cos r} = \frac{2n\cos i}{n\cos i + n'\cos r} \\
\frac{E_0''}{E_0} &= \frac{n\cos i - \frac{\mu}{\mu'}n'\cos r}{n\cos i + \frac{\mu}{\mu'}n'\cos r} = \frac{n\cos i - n'\cos r}{n\cos i + n'\cos r}
\end{aligned}
$$

   where we suppose, as it is legitimate for visible or near-visible light, that $\mu/\mu' \approx 1$;

2. a plane wave with the electric field parallel to the above surface:

$$
\begin{aligned}
\frac{E_0'}{E_0} &= \frac{2n\cos i}{\frac{\mu}{\mu'}n'\cos i + n\cos r} = \frac{2n\cos i}{n'\cos i + n\cos r} \\
\frac{E_0''}{E_0} &= \frac{\frac{\mu}{\mu'}n'\cos i - n\cos r}{\frac{\mu}{\mu'}n'\cos i + n\cos r} = \frac{n'\cos i - n\cos r}{n'\cos i + n\cos r}
\end{aligned}
$$

   whith the same approximation as above.

We note that in case of photon perpendicular to the surface, the following relations hold:

$$\frac{E'_0}{E_0} = \frac{2n}{n'+n} \qquad \frac{E''_0}{E_0} = \frac{n'-n}{n'+n}$$

where the sign convention for the *parallel* field has been adopted. This means that if $n' > n$ there is a phase inversion for the reflected wave.

Any incoming wave can be separated into one piece polarised parallel to the plane and one polarised perpendicular, and the two components treated accordingly.

To mantain the particle description of the photon, the probability to have a *refracted* (mechanism 107) or *reflected* (mechanism 106) photon must be calculated. The constraint is that the number of photons be conserved, and this can be imposed via the conservation of the energy flux at the boundary, as the number of photons is proportional to the energy. The energy current is given by the expression:

$$\vec{S} = \frac{1}{2}\frac{c}{4\pi}\sqrt{\mu\epsilon}\vec{E} \times \vec{H}^* = \frac{c}{8\pi}\sqrt{\frac{\epsilon}{\mu}}E_0^2\hat{k}$$

and the energy balance on a unit area of the boundary requires that:

$$\begin{aligned}
\vec{S} \cdot \vec{u} &= \vec{S}' \cdot \vec{u} - \vec{S}'' \cdot \vec{u} \\
S \cos i &= S' \cos r + S'' \cos i \\
\frac{c}{8\pi}\frac{1}{\mu}nE_0^2 \cos i &= \frac{c}{8\pi}\frac{1}{\mu'}n'E_0'^2 \cos r + \frac{c}{8\pi}\frac{1}{\mu}nE_0''^2 \cos i
\end{aligned}$$

If we set again $\mu/\mu' \approx 1$, then the transmission probability for the photon will be:

$$T = \left(\frac{E'_0}{E_0}\right)^2 \frac{n' \cos r}{n \cos i}$$

and the corresponding probability to be reflected will be $R = 1 - T$.

In case of reflection the relation between the incoming photon $(\vec{k}, \vec{e})$, the refracted one $(\vec{k}', \vec{e}\,')$ and the reflected one $(\vec{k}'', \vec{e}\,'')$ is given by the following relations:

$$\begin{aligned}
\vec{q} &= \vec{k} \times \vec{u} \\
e_{\parallel} &= \frac{\vec{e} \cdot \vec{u}}{|q|} \\
e_{\perp} &= \frac{\vec{e} \cdot \vec{q}}{|q|} \\
e'_{\parallel} &= e_{\parallel}\frac{2n \cos i}{n' \cos i + n \cos r} \\
e'_{\perp} &= e_{\perp}\frac{2n \cos i}{n \cos i + n' \cos r} \\
e''_{\parallel} &= \frac{n'}{n}e'_{\parallel} - e_{\parallel} \\
e''_{\perp} &= e'_{\perp} - e_{\perp}
\end{aligned}$$

After transmission or reflection of the photon, the polarisation vector is renormalised to 1. In the case where $\sin r = n \sin i/n' > 1$ then there cannot be a refracted wave, and in this case we have a total internal reflection according to the following formulas:

$$\begin{aligned}
\vec{k}'' &= \vec{k} - 2(\vec{k} \cdot \vec{u})\vec{u} \\
\vec{e}\,'' &= -\vec{e} + 2(\vec{e} \cdot \vec{u})\vec{u}
\end{aligned}$$

Case dielectric → metal

In this case the photon cannot be transmitted. So the probability for the photon to be absorbed by the metal is estimated according to the table provided by the user. If the photon is not absorbed, it is reflected.

3.3 Surface effects

In the case where the surface between two bodies is perfectly polished, then the normal provided by the program is the normal to the surface defined by the body boundary. This is indicated by the the value `POLISH`$= 1$ as returned by the `GUPLSH` function. When the value returned is < 1, then a random point is generated in a sphere of radius $1-$`POLISH`, and the corresponding vector is added to the normal. This new normal is accepted if the reflected wave is still inside the original volume.

4 Subroutines

`CALL` **GSCKOV** `(ITMED, NPCKOV, PPCKOV, ABSCO, EFFIC, RINDEX)`

`ITMED`	(`INTEGER`) tracking medium for which the optical properties are to be defined;
`NPCKOV`	(`INTEGER`) number of bins in the tables;
`PPCKOV`	(`REAL`) array containing `NPCKOV` values of the photon momentum in GeV;
`ABSCO`	(`REAL`) array containing `NPCKOV` values of the absorption length in centimeters in case of dielectric and of the boundary layer absorption probabilities in case of a metal;
`EFFIC`	(`REAL`) array containing `NPCKOV` values of the detection efficiency;
`RINDEX`	(`REAL`) array containing `NPCKOV` values of the refractive index for a dielectric, if `RINDEX(1) = 0` the material is a metal;

This routine declares a tracking medium either as a radiator or as a metal and stores the tables provided by the user. In the case of a metal the `RINDEX` array does not need to be of length `NPCKOV`, as long as it is set to 0. The user should call this routine if he wants to use Čerenkov photons. Please note that for the moment only `BOX`es, `TUBE`s, `CONE`s, `SPHE`res, `PCON`s, `PGON`s, `TRD2`s and `TRAP`s can be assigned optical properties due to the current limitations of the `GGPERP` routine described below.

`CALL` **GLISUR** `(X0, X1, MEDI0, MEDI1, U, PDOTU, IERR)`

`X0`	(`REAL`) current position (`X0(1)=`x, `X0(2)=`y, `X0(3)=`z) and direction (`X0(4)=`p_x, `X0(5)=`p_y, `X0(6)=`p_z) of the photon at the boundary of a volume;
`X1`	(`REAL`) position (`X1(1)=x, X1(2)=y, X1(3)=z`) beyond the boundary of the current volume, just inside the new volume along the direction of the photon;
`MEDI0`	(`INTEGER`) index of the current tracking medium;
`MEDI1`	(`INTEGER`) index of the tracking medium into which the photon is entering;
`U`	(`REAL`) array of three elements containing the normal to the surface to which the photon is approaching;
`PDOTU`	(`REAL`) $-\cos\theta$ where θ is the angle between the direction of the photon and the normal to the surface;
`IERR`	(`INTEGER`) error flag, `GGPERP` could not determine the normal to the surface if `IERR` $\neq$ `0`;

This routine simulates the surface profile between two media as seen by an approaching particle with coordinate and direction given by `X0`. The surface is identified by the arguments `MEDI0` and `MEDI1` which are the tracking medium indices of the region in which the track is presently and the one which it approaches, respectively. The input vector `X1` contains the coordinates of a point on the other side of the boundary from `X0` and lying in within medium `MEDI1`. The result is a unit vector `U` normal to the surface of reflection at `X0` and pointing into the medium from which the track is approaching.

The quality of the surface finish is given by the parameter returned by the user function `GUPLSH` (see below).

```
VALUE  = GUPLSH  (MEDIO, MEDI1)
```

This function must be supplied by the user. It returns a value between 0 and 1 which decsribes the quality of the surface finish between MEDIO and MEDI1. The value 0 means maximum roughness with effective plane of reflection distributed as $\cos\alpha$ where α is the angle between the unit normal to the *effective* plane of reflection and the normal to the nominal medium boundary at X0. The value 1 means perfect smoothness. In between the surface is modelled as a bell-shaped distribution in α with limits given by:

$$\sin\alpha = \pm(1 - \mathtt{GUPLSH})$$

At the interface between two media the routine is called to evaluate the surface. The default version in GEANT returns 1, i.e. a perfectly polished surface is assumed. When GUPLSH = 0 the distribution of the normal to the surface is $\approx \cos\theta$.

MEDIO	(INTEGER) index of the current tracking medium;
MEDI1	(INTEGER) index of the tracking medium into which the photon is entering;

```
CALL GGCKOV
```

This routine handles the generation of Čerenkov photons and is called from GTHADR, GTMUON and GTELEC in radiative materials for which the optical characteristics have been defined via the routine GSCKOV.

```
CALL GSKPHO  (IPHO)
```

IPHO	(INTEGER) number of the Čerenkov photon to store on the stack to be tracked. If IPHO = 0 all the generated photons will be put on the stack to be tracked.

This routines takes the Čerenkov photon IPHO generated during the current step and stores it in the stack for subsequent tracking. This routine performs for Čerenkov photons the same function that the GSKING routine performs for all the other particles. The generated photons are stored in the common /GCKIN2/ ([ZZZZ010]).

```
CALL GTCKOV
```

This routine is called to track Čerenkov photons. The user routine GUSTEP is called at every step of tracking. When ISTOP = 2 the photon has been absorbed. If DESTEP $\neq 0$ then the photon has been detected.

```
CALL GGPERP  (X, U, IERR)
```

X	(REAL) array of dimension 3 containing the current position of the track in the MARS;
U	(REAL) array of dimension 3 containing on exit the normal to the surface of the current volume at the point X;
IERR	(INTEGER) error flag: if IERR $\neq 0$ GGPERP failed to find the normal to the surface of the current volume.

This routine solves the general problem of calculating the unit vector normal to the surface of the current volume at the point X. The result is returned in the array U. X is assumed to be outside the current volume and near a boundary (within EPSIL). The current volume is indicated by the common /GCVOLU/. U points from inside to outside in that neighbourhood. If X is within EPSIL of more than one boundary (i.e. in a corner) an arbitrary choice is made. If X is inside the current volume or if the current volume is not handled by the routine, the routine returns with IERR=1, otherwise IERR=0. At the moment the routine only handles BOXes, TUBEs, SPHEres and CONEs.

5 Processes involving Čerenkov photons

The process of generating a Čerenkov photon is called `CKOV` and corresponds to the process value 105 (variable `LMEC` in common `/GCTRAK/`). This process is activated only in a radiator defined via the routine `GSCKOV`.

The process of photon absorption (name `LABS`, code 101) is controlled by the `LABS` `FFREAD` data record. By default the process is activated for all the materials for which optical properties have been defined.

The action taken at the boundary is identified by the process name `LREF`, code 102.

At a boundary a photon can be either reflected (name `REFL`, code 106) or refracted (name `REFR`, code 107).

Origin : R.Brun, M.Hansroul
Revision : F.Carminati, G.R.Lynch
Documentation : M.Hansroul, G.R.Lynch

Submitted: 17.06.85
Revised: 16.12.93

Gaussian multiple scattering

1 Subroutines

CALL **GMGAUS** (BETA2,DIN*)

BETA2 (REAL) β^2;

DIN (REAL) array of 3 containing the direction of the scattered particle with respect to the incoming direction.

This routine is called by GTELEC, GTHADR, GTMUON and GTHION via GMULTS when the variable IMULS, controlled by the data record MULS is equal to 3.

2 Method

2.1 Approximations in use

A charged particle traversing a medium is deflected by many scatters, mostly at small angle. These scatters are due to the interaction with the Coulomb field of the nuclei, and to a lesser degree, to the electron field, hence the name of Coulomb scattering. For hadronic projectiles, however, the strong interaction contributes to multiple scattering. Multiple scattering is well described by Molière theory [37]. Molière multiple scattering theory is used by default in GEANT (see [PHYS325]). We define $\theta_0 = \theta^{rms}_{plane} = \theta^{rms}_{space}/\sqrt{2}$ as the r.m.s. of the angle between the directions projected on a plane of a particle before and after traversing a thickness t of absorber. In this case a simple form for the multiple Coulomb scattering of singly charged particles which is widely used is:

$$\theta_0 \approx \frac{15MeV}{E\beta^2}\sqrt{\frac{t}{X_0}}$$

where X_0 is the radiation length of the absorber. This form was proposed by [38], [39]. It can introduce large errors because it ignores significant dependences from pathlength and Z. To compensate for this, a similar formula was proposed [40], [41]:

$$\theta_0 \approx \frac{14.1MeV}{E\beta^2}\sqrt{\frac{t}{X_0}}\left[1 + 0.038\log\left(\frac{t}{X_0}\right)\right] \tag{1}$$

A form taking into account the β and z dependence of the particle has been proposed by [42]:

$$\theta_0 = \frac{S_2}{E\beta^2}\sqrt{\frac{t}{X_0}}\left[1 + \epsilon\log_{10}\left(\frac{tZ^2_{inc}}{X_0\beta^2}\right)\right] \tag{2}$$

The problem with the formulae (1) and (2) is that if the distance t in the absorber is travelled in two steps $t = t_1 + t_2$, the r.m.s. angle $\theta_0(t) = \theta_0(t_1 + t_2) \neq \sqrt{\theta_0^2(t_1) + \theta_0^2(t_2)}$, limiting their use in a MonteCarlo like GEANT.

Much of the difficulty in approximating multiple Coulomb scattering in terms of the radiation length is that the number of radiation lengths is a poor measure of the scattering. A better expression, which is used in GEANT has been proposed by [42]:

$$\theta_0^2 = \frac{\chi_c^2}{1+F^2}\left[\frac{1+\nu}{\nu}\log(1+\nu) - 1\right] \tag{3}$$

where

$$\begin{aligned}
\nu &= \frac{\Omega_0}{2(1-F)} \\
F &= \text{fraction of the tracks in the sample} \\
\Omega_0 &= \frac{\chi_c^2}{1.167\chi_\alpha^2} \qquad \text{is the mean number of scatters}
\end{aligned}$$

χ_c^2 and χ_α^2 are parameters of the Molière theory, for which the reader is invited to see [PHYS325], and $1.167 \approx \exp(2\gamma - 1)$ where γ is the Euler's constant $= 0.57721\ldots$. This form, which is the result of empirical fits, is derived from the screened Rutherford cross section, which has the form $2\chi_\alpha^2/(\chi_\alpha^2 + \theta_0^2)$. For F anywhere from 0.9 to 0.995 this expression represents Molière scattering to better than 2% for $10 < \Omega_0 < 10^8$, which covers singly charged relativistic particles ($\beta \approx 1$) in the range $10^{-3} < t/X_0 < 100$. In GEANT we have adopted the values of $F = 0.98$ and $\Omega_0 = 40,000$ scatters, obtaining the following formula:

$$\theta_0 = 2.557\chi_{cc}\frac{\sqrt{t}}{E\beta^2}$$

where χ_{cc} is another parameter of the Molière theory for which the reader is referred to [PHYS325]. This algorithm is implemented by the routine GMGAUS.

Origin : M.S. Dixit
Revision : G.Lynch
Documentation : F.Carminati

Submitted: 17.06.85
Revised: 16.12.93

Molière scattering

1 Subroutines

1.1 Initialisation

CALL **GMULOF**

GMULOF calculates the table of maximum steps corresponding to the energies in the array ELOW for electrons and muons. It is called during initialisation by GPHYSI. For hadrons and ions the computation is performed during tracking in the routines GTHADR and GTHION.

CALL **GMOLI** (AC,ZC,WMAT,NLM,DENS,OMC*,CHC*)

AC	REAL array of dimenstion NLM containing the mass numbers;
ZC	REAL array of dimenstion NLM containing the atomic numbers;
WMAT	REAL array containing the proportions by weight of the component of a mixture/compound. 1 in case of a single component;
NLM	INTEGER number of the elements in the mixture/compound;
DENS	REAL density in g cm^{-3};
OMC	REAL b_c constant of Molière theory;
CHC	REAL χ_{cc} constant of Molière theory.

GMOLI calculates the two material-dependent constants (OMC and CHC). These are needed during initialisation by GMULOF and during tracking to sample the scattering angle, as explained below. It is called at initialisation time by the routine GPROBI which initialises some material constants and computes the probabilities for various processes.

1.2 Sampling of the multiple scattering angle

CALL **GMULTS**

GMULTS is the steering routine for the sampling of the multiple scattering angle when the variable IMULTS in common /GCPHYS/ is 1 or 2. This variable can be controlled by the MULT data record. The routine decides if the material-dependent constant OMC computed by GMOLI during initialisation has to be corrected due to its dependence on β and Z_{inc}, and selects Molière theory (GMOLIE) or single Coulomb scattering (GMCOUL). It updates the direction of the particle in the array VECT (COMMON GCTRAK). It is called by the tracking routines GTELEC, GTHADR, GTHION and GTMUON.

CALL **GMOLIO** (AC,ZC,WMAT,NLM,DENS,BETA2,CHARG2,OMC*)

BETA2	REAL β^2 of the particle;
CHARG2	REAL charge squared of the particle in electron charge unit (Z^2_{inc} in the following).

All the other arguments are routine GMOLI. GMOLIO re-evaluates the material-dependent constant OMC. It is called by GMULTS, if needed.

CALL **GMOLIE** (OMEGA,BETA2,DIN*)

OMEGA REAL Ω_0 of the Molière theory;
BETA2 REAL β^2 of the particle;
DIN REAL array of dimension 3 containing the new direction cosines with respect to the original direction of the particle.

GMOLIE samples the multiple scattering angle according to Molière theory, as explained in the next section. It is called by the multiple scattering steering routine GMULTS,

```
CALL GMOL4 (Y*,X,VAL,ARG,EPS,IER*)
```

Y REAL value of the interpolated function;
X REAL value at which the function should be interpolated;
VAL REAL array of four values of the function to interpolate;
ARG REAL array of four values of the argument of the function to which the values contained in VAL correspond;
EPS REAL required precision for the interpolation;
IER INTEGER error flag;

GMOL4 inverts the integral of Molière distribution function using four-points continued-fraction interpolation. It is called by GMOLIE.

2 Method

2.1 Introduction

When traversing ordinary matter, elementary particles undergo deflection from their originary trajectory due to the interaction with the atoms. This effect is rather large for charged particles, which are deflected by the electric field of nuclei and electrons via a large number of small elastic *collisions*. To simulate precisely the transport of particles in matter, it is important to provide a precise description of this effect. Considerable effort has gone over the years into the formulation of a theory of Coulomb multiple scattering. In GEANT we follow the theory originally formulated by Molière [43] [44] and then elaborated by Bethe [37]. A review of this theory can be found in [45].

It should be noted that this is not the only theory which describes multiple Coulomb scattering, and a thorough analysis of the problem has been performed by Goudsmit and Saunderson [46] [47].

A complete account of the Molière theory is outside the scope of this manual. For more information the interested reader could refer to the works cited above and to the EGS4 manual [15], from which the notation used in the present chapter has been derived.

The principal limitations of Molière theory are the following:

1. the angular deflection is *small*. Effectively this constraint provides the upper limit for the step-size;
2. the theory is a *multiple scattering* theory; that is, many atomic collisions participate in causing the incident particle to be deflected. In some implementations, this constraints provides a lower limit on the step size. In GEANT, when the number of scatters is below the limit of applicability of Molière single Coulomb scatterings are simulated, so that there is no effective minimum step size below which multiple scattering is *switched off*;
3. the theory applies only in semi-infinite homogeneous media. This constraint calls for a very careful handling of particles near to a medium's boundary;
4. there is no energy loss built into the theory. This again introduces the necessity to take small steps;

In Molière theory corrected for finite angle scattering ($\sin\theta \neq \theta$) as described by Bethe [37, 45], the angular distribution is given by:

$$f(\theta)\,\theta d\theta \quad = \quad \sqrt{\frac{\sin\theta}{\theta}}\, f_r(\eta)\,\eta d\eta, \tag{1}$$

where for $f_r(\eta)$ we use the first three terms of Bethe's expansion:

$$\begin{aligned} f_r(\eta) &= f_r^{(0)}(\eta) + f_r^{(1)}(\eta)B^{-1} + f_r^{(2)}(\eta)B^{-2} \\ \eta &= \frac{\theta}{\chi_c\sqrt{B}} \end{aligned} \tag{2}$$

η is called the *reduced angle*.

2.2 Calculation of the reduced angle

In formula (2) above χ_c the *critical scattering angle* defined as:

$$\chi_c^2 = \frac{4\pi e^4 Z_s \rho N_{Av} Z_{inc}^2 t}{W c^2 p^2 \beta^2} = \chi_{cc}^2 Z_{inc}^2 \frac{t}{E^2\beta^4} \tag{3}$$

with

Z_{inc}	incident particle charge
Z_s	defined below
N_{Av}	Avogadro's number
ρ	density
W	$\sum_{i=1}^{N} n_i A_i$ molecular weight
E	particle energy in MeV
r_0	$e^2/(mc^2)$ classical electron radius
t	total path length in the scatterer (and not its thickness).

The variable χ_c is factorised into two parts. The first part, χ_{cc}, is a constant of the medium for a given incident particle. The second part depends on the incident particle energy and on the path length in the medium.

$$\chi_{cc}^2 = \frac{4\pi e^4 Z_s \rho N_{Av}}{W} = \frac{4\pi c^4 r_0^2 m_e^2 Z_s \rho N_{Av}}{W} \approx (0.39612\,10^{-3})^2 Z_s \frac{\rho}{W}\,[\,GeV^2\ cm^{-1}] \tag{4}$$

The variable `CHC` or `CHCMOL` stored in `Q(JPROB+25)` corresponds to χ_{cc}. B is defined by the equation

$$B - \ln B = \ln \Omega_0, \tag{5}$$

where Ω_0 can be interpreted as the number of collisions in the step:

$$\Omega_0 = \frac{\chi_c^2}{e^{2\gamma-1}\chi_\alpha^2} = b_c Z_{inc}^2 \frac{t}{\beta^2} \tag{6}$$

Here γ is Euler's constant ($\gamma = 0.57721\ldots$) and χ_α the atomic electron screening angle. For a single element this is given by:

$$\chi_\alpha^2 = \left(\frac{\lambda_0}{2\pi r_{TF}}\right)^2 \left[1.13 + 3.76\left(\frac{\alpha Z Z_{inc}}{\beta}\right)^2\right]$$

where

$$\begin{aligned} \frac{\lambda_0}{2\pi} &= \hbar/p \quad \text{Compton wavelength of the electron} \\ r_{TF} &= 0.885 a_0 Z^{-\frac{1}{3}} \quad \text{Thomas-Fermi radius of the atom} \\ a_0 &= \frac{\hbar^2}{me^2} \quad \text{Bohr's radius} \\ \alpha &= 1/137.035\ldots \quad \text{the fine-structure constant} \end{aligned}$$

so that we have

$$\chi_\alpha^2 = \frac{m^2 e^4 1.13}{p^2\hbar^2(0.885)^2} Z^{\frac{2}{3}} \left[1 + 3.34\left(\frac{\alpha Z Z_{inc}}{\beta}\right)^2\right].$$

For a compound/mixture the following rule holds:

$$\begin{aligned}
\ln(\chi_\alpha^2) &= \frac{\sum_{i=1}^{N} n_i Z_i(Z_i+1)\ln(\chi_{\alpha i}^2)}{\sum_{i=1}^{N} n_i Z_i(Z_i+1)} = \frac{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)\ln(\chi_{\alpha i}^2)}{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)} \\
&= \frac{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)\ln\left\{\frac{m^2e^4 1.13}{p^2\hbar^2(0.885)^2} Z_i^{\frac{2}{3}}\left[1+3.34\left(\frac{\alpha Z_i Z_{inc}}{\beta}\right)^2\right]\right\}}{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)} \\
&= \ln\left(\frac{m^2e^4 1.13}{p^2\hbar^2(0.885)^2}\right) - \frac{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)\ln Z_i^{-\frac{2}{3}}}{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)} \\
&\quad + \frac{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)\left[1+3.34\left(\frac{\alpha Z_i Z_{inc}}{\beta}\right)^2\right]}{\sum_{i=1}^{N} \frac{p_i}{A_i} Z_i(Z_i+1)}
\end{aligned}$$

To understand the transformations in the above formulae, the following should be noted. Let n_i be the number of atoms of type i in a compound/mixture, that is the number of moles of element i in a mole of material, and p_i the proportion by weight of that element. We have the following relation:

$$p_i = \frac{n_i A_i}{\sum_{j=1}^{N} n_j A_j} \qquad \text{that is} \qquad n_i = \left(\sum_{j=1}^{N} n_j A_j\right)\frac{p_i}{A_i} = W\frac{p_i}{A_i}$$

so that we can simplify expressions in the following way:

$$\frac{\sum_{i=1}^{N} n_i \ldots}{W} = \sum_{i=1}^{N}(p_i/A_i)\ldots$$

If now we set

$$\begin{aligned}
Z_s &= \sum_{i=1}^{N} n_i Z_i(Z_i+1) && = W\sum_{i=1}^{N}\frac{p_i}{A_i}Z_i(Z_i+1) && = WZ_s' \\
Z_E &= \sum_{i=1}^{N} n_i Z_i(Z_i+1)\ln Z_i^{-2/3} && = W\sum\frac{p_i}{A_i}Z_i(Z_i+1)\ln Z_i^{-2/3} && = WZ_E' \\
Z_x &= \sum_{i=1}^{N} n_i Z_i(Z_i+1)\ln\left[1+3.34\left(\frac{\alpha Z_i Z_{inc}}{\beta}\right)^2\right] \\
&= W\sum_{i=1}^{N}\frac{p_i}{A_i}Z_i(Z_i+1)\ln\left[1+3.34\left(\frac{\alpha Z_i Z_{inc}}{\beta}\right)^2\right] && && = WZ_x'
\end{aligned}$$

we can write:

$$
\begin{aligned}
\ln(\chi_\alpha^2) &= \ln\left(\frac{m^2e^4 1.13}{p^2\hbar^2(0.885)^2}\right) + \frac{Z_x - Z_E}{Z_s} \\
\chi_\alpha^2 &= \frac{m^2e^4 1.13}{p^2\hbar^2(0.885)^2} e^{(Z_x - Z_E)/Z_s} = \frac{m^2e^4 1.13}{p^2\hbar^2(0.885)^2} e^{(Z'_x - Z'_E)/Z'_s}
\end{aligned}
$$

and finally:

$$
\begin{aligned}
\Omega_0 &= \frac{\chi_c^2}{e^{2\gamma-1}\chi_\alpha^2} \\
&= \frac{1}{1.167}\,\frac{4\pi e^4 Z_s \rho N_{Av} Z_{inc}^2 t}{W c^2 p^2 \beta^2}\,\frac{p^2\hbar^2(0.885)^2}{1.13 m^2 e^4} e^{-(Z'_x - Z'_E)/Z'_s} \\
&= b_c Z_{inc}^2 \frac{t}{\beta^2} \\
b_c &= \frac{4\pi N_{Av}\hbar^2}{m^2c^2}\,\frac{(0.885)^2}{1.13\times 1.167}\rho Z'_s e^{(Z'_E - Z'_x)/Z'_s} \approx 6702.33\rho Z'_s e^{(Z'_E - Z'_x)/Z'_s}
\end{aligned}
$$

The quantity b_c is calculated during initialisation by the routine `GMOLI` setting $\beta = 1$ and $Z_{inc} = 1$. This variable is called `OMC` or `OMCMOL` and it is stored in `Q(JPROB+21)`. It has, via the atomic electron screening angle χ_α^2, a small dependence on β and Z_{inc} in the term Z_x. The quantity $b_c\rho Z'_s$ is re-evaluated during tracking by the routine `GMOLIO`, called by `GMULTS`, only when necessary.
The dependence of Z_x on β and Z_{inc} is via a term of the form: $1 + 3.34\left(\alpha Z Z_{inc}\beta^{-1}\right)^2$, so it is not necessary to recalculate it as long as $1 + 3.34\left(\alpha Z Z_{inc}\beta^{-1}\right)^2 \approx 1 + 3.34\left(\alpha Z\right)^2$ that is:

$$
\begin{aligned}
3.34(Z\alpha)^2\left[\left(\frac{Z_{inc}}{\beta}\right)^2 - 1\right] &\ll 1 \qquad \Rightarrow \quad Z^2\frac{Z_{inc}^2 - \beta^2}{\beta^2} \quad \ll \quad 3.34\alpha^{-2} \approx 5500 \\
Z^2(Z_{inc}^2 - \beta^2) &\leq 50\beta^2
\end{aligned}
$$

For mixtures/compounds this condition should be checked for every component and this would be unacceptable from the point of view of computer time. So we simply make the condition more restrictive multiplying the first term by the number of elements in the mixture/compound.

2.3 Sampling of the distribution function

The component distribution functions (2) are given by

$$
\begin{aligned}
f_r^{(0)}(\eta) &= 2e^{-\eta^2} = 2D_0(1,1,-\eta^2) \\
f_r^{(1)}(\eta) &= 2D_1(2,1,-\eta^2) \\
f_r^{(2)}(\eta) &= D_2(3,1,-\eta^2),
\end{aligned}
$$

where $D_n(a,b,z) = d^n[\Gamma(a)M(a,b,z)]/da^n$, with $M(a,b,z)$ the Kummers hypergeometric function.
Integrals of the functions $f_r^{(1)}$ and $f_r^{(2)}$ needed for the Monte Carlo can be written down directly in terms of the D functions

$$
\begin{aligned}
\int_\eta^\infty \eta f_r^{(1)}\,\eta d\eta &= D_1(2,1,-\eta^2) - D_1(1,1,-\eta^2) - D_0(1,1,-\eta^2) \\
\int_\eta^\infty \eta f_r^{(2)}\,\eta d\eta &= \frac{1}{2}D_2(3,1,-\eta^2) - D_2(2,1,-\eta^2) - D_1(2,1,-\eta^2)
\end{aligned}
$$

We note that the first term is just the Gaussian part and it dominates for large B, that is, for large number of scatters, as it has to be expected. Recalling the definition of η in (2) we can say that the half-width of the Gaussian part of the Molière distribution is $\sigma^2 = \chi_c^2 B/2$. Routine `GMOLIE` performs the sampling of the Molière distribution via the following steps:

1. the value of B is calculated recursively. If we set $f(B) = B$ with $f(B) = \ln \Omega_0 + \ln B$ the relation used is $B_n = B_{n-1} + \Delta B = B_{n-1} + (f(B_{n-1}) - B_{n-1})/(1 - 1/B_{n-1})$. Convergence is assumed when $|\Delta B| \leq 10^{-4}$.

2. a random number r_1 is sampled and a table of four values $F_r(\eta_i)$ is built with $i = j-2, j-1, j, j+1$ and $F_r(\eta_j) \geq r_1$. F is the distribution function derived from (2):

$$\begin{aligned} F_r(\eta) &= \int_0^\eta f_r(t)\, dt = \int_0^\eta f_r^{(0)}(t) + f_r^{(1)}(t)B^{-1} + f_r^{(2)}(t)B^{-2}\, dt \\ &= \int_0^\eta f_r^{(0)}(t)\, dt + B^{-1}\int_0^\eta f_r^{(1)}(t)\, dt + B^{-2}\int_0^\eta f_r^{(2)}(t)\, dt \\ &= F_r^{(0)}(\eta) + B^{-1}F_r^{(1)}(\eta) + B^{-2}F_r^{(2)}(\eta) \end{aligned}$$

   40 values of the functions $F_r^{(n)}$ are tabulated.

3. using a four-points continued-fraction interpolation method (`GMOL4`) a table of η_i^2 is interpolated using the values of r_1 and F tabulated in the previous step. This is similar to the inversion of the distribution function, but instead of obtaining directly the random variable, we interpolate a table of its squares. This provides a better fit to the inverse function;

4. the value of $\theta = \chi_c \sqrt{B\eta^2}$ is calculated;

5. a random number r_2 is extracted, and the rejection function $g(\theta) = \sqrt{\sin\theta/\theta}$ is computed. If $r_2 > g(\theta)$ resampling begins from step 2, otherwise the value is accepted.

The Molière distribution gives the space scattering angle. A similar expression may be written for the lateral displacement of the scattered particles. However, the problem of joint angle lateral displacement in the Molière approximation has not been solved, and, for small step size, lateral displacement is of second order and may be neglected. This introduces a problem when large step sizes are taken. The step size can be artificially limited via the use of the variable `STEMAX`, which is an argument to the routine `GSTMED`. For more information on this point the user is invited to consult chapter `[CONS200]`.

3 Path and step length

3.1 Restrictions on the step length

Restrictions on the length of the step arise from:

1. the number of scatters $\Omega_0 \geq 20$ to stay within the multiple scattering regime. When $\Omega_0 < 20$, an appropriate number of *single* scatterings is used. See routine `GMCOUL [PHYS328]`.

2. $\chi_c^2 B \leq 1$ i.e. the width of the Gaussian part of the distribution should be less than one radian. This condition induces a maximum step length for the multiple scattering called t_{Bethe}. In order to find this value we write the limiting condition as $\chi_c^2(t_{Bethe})B(t_{Bethe}) = 1$, that is $B(t_{Bethe}) = 1/\chi_c^2(t_{Bethe})$. Now using equation (5) we take the exponential of both members $B(t_{Bethe})\Omega_0(t_{Bethe}) = \exp(1/\chi_c^2(t_{Bethe}))$. Dividing the two last equalities we obtain:

$$\begin{aligned} \Omega_0(t_{Bethe}) = \frac{b_c Z_{inc}^2}{\beta^2} t_{Bethe} &= \chi_c^2(t_{Bethe})\exp(1/\chi_c^2(t_{Bethe})) \\ &= \frac{\chi_{cc}^2 Z_{inc}^2 t_{Bethe} \exp\left[(E^2\beta^4)/(\chi_{cc}^2 Z_{inc}^2 t_{Bethe})\right]}{E^2\beta^4} \\ \exp\left[(E^2\beta^4)/(\chi_{cc}^2 Z_{inc}^2 t_{Bethe})\right] &= \frac{b_c E^2\beta^2}{\chi_{cc}^2} \\ t_{Bethe} &= \frac{E^2\beta^4}{\chi_{cc}^2\ Z_{inc}^2\ \ln\left[b_c E^2\beta^2/\chi_{cc}^2\right]} \end{aligned}$$

For electrons and muons this constraint on the step-length is tabulated at initialisation time in the routine `GMULOF [PHYS201]`. For hadrons this formula can be approximated as:

$$t_{Gauss} \approx \left(\frac{1}{14.1\,10^{-3}}\frac{E^2\beta}{Z_{inc}}\right)^2 X_0,$$

where E is in GeV and X_0 is the radiation length in centimeters and the formula has been taken from the Gaussian approximation to multiple scattering (see `[PHYS320]`). This condition is more restrictive, because it is equivalent to require that the width of the Gaussian part of the distribution be less than 0.5, but it has been found that the two conditions are numerically equivalent;

3. limitation due to the path length correction algorithm used (see below).

3.2 Path Length Correction

A path length correction may be applied in an approximate manner. We have from the Fermi-Eyges theory [15]

$$t = S + \frac{1}{2}\int_0^t \bar{\theta}^2(\bar{t})\,dt \tag{7}$$

where

$\bar{\theta}^2(\bar{t})$	the mean square angle of scattering;
S	step size along a straight line;
t	actual path length.

We have further:

$$\bar{\theta}^2(\bar{t}) = \left(0.0212\frac{Z_{inc}}{p\beta}\right)^2 \frac{t}{X_0} \tag{8}$$

X_0 is the radiation length. From (7) and (8) we get

$$S = t - Kt^2 \qquad \text{with} \qquad K = 1.12\times 10^{-4}\,\frac{Z_{inc}^2}{p^2\beta^2 X_0} \tag{9}$$

Equation (9) may be used to make the path length correction. Solving equation (9) with respect to t implies that, in order to obtain real solutions:

$$S \leq \frac{1}{4K} \qquad \text{i.e.} \qquad S \leq 2232\,\frac{X_0 p^2\beta^2}{Z_{inc}^2} \tag{10}$$

This condition provides an additional constraint to the maximum step length for multiple scattering (variable `TMXCOR` in routine `GMULOF`). The corrected step can be approximated as:

$$t \approx S(1+KS) = S(1+\texttt{CORR})$$

where $\texttt{CORR} \leq 0.25$ due to condition (10).

Origin : G.Lynch
Revision :
Documentation : F.Carminati

Submitted: 23.08.93
Revised: 16.12.93

Plural scattering

1 Subroutines

CALL **GMCOUL** (OMEGA,DIN*)

OMEGA (REAL) Ω_0 of the Molière theory;

DIN (REAL) array of dimension 3 containing the direction of the scattered particle with respect to the incoming direction;

Generates single scatters in small angle approximation returning the new direction with respect to the old one. It is called by GMULTS when $\Omega_0 \leq 20$.

2 Method

In the Molière theory the average number of Coulomb scatters for a charged particle in a step is expressed by the parameter Ω_0 (see [PHYS325]). When $\Omega_0 \leq 20$, the Molière theory is not applicable any more, even if it has been noted [48] that it still gives reasonable results down to its *mathematical limit* $\Omega_0 = e$. The range $1 < \Omega_0 \leq 20$ is called the *plural scattering* regime. An interesting study of this regime can be found in [49].

In GEANT, when $\Omega_0 \leq 20$, a direct simulation method is used for the scattering angle. The number of scatters is distributed according to a Poissonian law with average $\Omega = k\Omega_0 = e^{2\gamma-1}\Omega_0 \approx 1.167\Omega_0$ with $\gamma = 0.57721\ldots$ Euler's constant. Using the customary notations for the probability distribution function for small angle ($\sin\theta \approx \theta$) single scattering, we can write:

$$f(\theta)\theta d\theta = \frac{d\sigma}{\theta d\theta}\frac{1}{\sigma}\theta d\theta \tag{1}$$

where σ is the cross section for single elastic scattering. We use as cross section the one reported by Molière [43] [44]:

$$\frac{d\sigma}{\theta d\theta} = 2\pi\left(\frac{2ZZ_{inc}e^2}{pv}\right)^2\frac{1}{(\theta^2+\chi_\alpha^2)^2}$$

This is the classical Rutherford cross section corrected by the screening angle χ_α. This angle is described by Molière as a correction to the Born approximation used to derive the quantum mechanical form of the Rutherford cross section. We have then:

$$\sigma = \int_0^\infty \frac{d\sigma}{\theta d\theta}\theta d\theta = 2\pi\left(\frac{2ZZ_{inc}e^2}{pv}\right)^2\int_0^\infty\frac{\theta d\theta}{(\theta^2+\chi_\alpha^2)^2} = \pi\left(\frac{2ZZ_{inc}e^2}{pv}\right)^2\frac{1}{\chi_\alpha^2}$$

and so equation (1) becomes:

$$f(\theta)\theta d\theta = \chi_\alpha^2\frac{2\theta d\theta}{(\theta^2+\chi_\alpha^2)^2} = \frac{2\Theta d\Theta}{(1+\Theta^2)^2}$$

where we have set $\Theta = \theta/\chi_\alpha$. To sample from this distribution we calculate the inverse of the distribution function:

$$\eta = \int_0^\Theta \frac{2t\,dt}{(1+t^2)^2} = 1 - \frac{1}{1+\Theta^2} \qquad \Rightarrow \qquad \Theta = \sqrt{\frac{1}{1-\eta} - 1}$$

where η is a number uniformly distributed between 0 and 1. If we observe that also $1-\eta$ is uniformly distributed between 0 and 1 and we remember the definition of Θ, we obtain:

$$\theta = \sqrt{\chi_\alpha^2 \left(\frac{1}{\eta} - 1\right)}$$

To calculate χ_α^2 we observe that:

$$\begin{aligned}
\Omega_0 &= \frac{\chi_c^2}{k\chi_\alpha^2} \\
\chi_\alpha^2 &= \frac{\chi_c^2}{k\Omega_0} = \frac{\chi_{cc}^2 Z_{inc}^2 t/E^2\beta^4}{kb_c Z_{inc}^2 t/\beta^2} = \frac{\chi_{cc}^2}{kb_c p^2 c^2}
\end{aligned}$$

where we have used the notations of [PHYS325] and $k = 1.167$.

Origin : L.Urbán **Submitted:** 26.10.84
Revision : **Revised:** 16.12.93
Documentation :

Ionisation processes induced by $e^{\pm}$

1 Subroutines

CALL **GDRELA**

GDRELA initialises the ionisation energy loss tables for different materials for muons, electrons and positrons and other particles. 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 (4) below) 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 with the energy loss coming from bremsstrahlung. For energy loss of electrons and positrons, it calls GDRELE using the following pointers:

JMA = LQ(JMATE-I)	pointer to the Ith material
JEL1 = LQ(JMA-1)	pointer for dE/dx for electrons
JEL1+90	pointer for dE/dx for positron

GDRELA is called at initialisation time by GPHYSI.

CALL **GDRELE** (EEL,CHARGE,JMA,DEDX)

GDRELE computes the ionisation energy loss (DEDX) for electrons (CHARGE = -1) and positrons (CHARGE = +1) with kinetic energy EEL in the material indicated by I in JMA = LQ(JMATE-I). It is called by the routine GDRELA.

CALL **GDRSGA**

CDRSGA calculates the total cross-section in all materials for delta rays for Möller (e^-e^-) and Bhabha (e^-e^+) scattering and for muons. It tabulates the mean free path, $\lambda = 1/\Sigma$ (in cm) as a function of the medium and the energy. The energy binning is set within the array ELOW (common /GCMULO/) in the routine GPHYSI. The following pointers are used (see JMATE data structure):

JMA = LQ(JMATE-I)	pointer to the Ith material
JDRAY = LQ(JMA-11)	pointer to δ-ray cross-sections
JDRAY	pointer for electrons
JDRAY+90	pointer for positrons
JDRAY+180	pointer for μ^+/μ^-.

The routine is called during initialisation by GPHYSI.

2 Method

Let:

$$\frac{d\sigma(E,T)}{dT}$$

be the differential cross-section for the ejection of an electron with kinetic energy T by an incident $e^{\pm}$ of total energy E moving in a medium of density ρ.

The variable DCUTE in common /GCCUTS/ is the kinetic energy cut-off for the generated δ-rays. Below this threshold the soft electrons ejected are simulated as continuous energy loss by the incident $e^{\pm}$, and above it they are explicitly generated.

The mean value of the energy lost by the incident $e^{\pm}$ to the soft δ-rays is:

$$E_{soft}(E, \mathtt{DCUTE}) = \int_0^{\mathtt{DCUTE}} \frac{d\sigma(E,T)}{dT} T \, dT \tag{1}$$

whereas the total cross-section for the ejection of an electron of energy $T > \mathtt{DCUTE}$ is:

$$\sigma(E, \mathtt{DCUTE}) = \int_{\mathtt{DCUTE}}^{T_{Max}} \frac{d\sigma(E,T)}{dT} \, dT \tag{2}$$

where T_{Max} is the maximum energy transferable to the free electron:

$$T_{Max} = \begin{cases} E - m & for \;\; \mathrm{e}^+ \\ (E-m)/2 & for \;\; \mathrm{e}^-, \end{cases} \tag{3}$$

where m is the electron mass. The method of calculation of the continuous energy loss and the total cross-section are explained below. [PHYS331] deals with the explicit simulation of the δ-rays.

2.1 Continuous energy loss

The integration of (1) leads to the Berger-Seltzer formulae [50, 51, 52, 12, 53, 54]:

$$\frac{dE}{dx} = \frac{2\pi r_0^2 mn}{\beta^2} \left[\ln \frac{2(\tau+2)}{(I/m)^2} + F^{\pm}(\tau, \Delta) - \delta \right], \tag{4}$$

where

γ	$\frac{E}{m}$
β^2	$1 - \frac{1}{\gamma^2}$
τ	$\gamma - 1$
τ_c	$\frac{\mathtt{DCUTE}}{m}$
DCUTE	energy cut for $e^{\pm}$
τ_{max}	maximum possible energy transfer in e$^-$mass: τ for e$^+$, $\tau/2$ for e$^-$
Δ	$\min(\tau_c, \tau_{max})$
n	electron density of the medium
I	average mean ionisation energy
δ	density effect correction.

The functions $F^{\pm}$ are given by

$$F^+(\tau, \Delta) = \ln(\tau\Delta) - \frac{\Delta^2}{\tau}\left[\tau + 2\Delta - \frac{3\Delta^2 y}{2} - \left(\Delta - \frac{\Delta^3}{3}\right) y^2 - \left(\frac{\Delta^2}{2} - \tau\frac{\Delta^3}{3} + \frac{\Delta^4}{4}\right) y^3 \right] \tag{5}$$

$$F^-(\tau, \Delta) = -1 - \beta^2 + \ln\left[(\tau - \Delta)\Delta\right] + \frac{\tau}{\tau - \Delta} + \frac{\left[\frac{\Delta^2}{2} + (2\tau+1)\ln\left(1 - \frac{\Delta}{\tau}\right)\right]}{\gamma^2}, \tag{6}$$

where $y = 1/(\gamma + 1)$.
The density effect correction is calculated as in [53]:

$$\delta = \begin{cases} 0 & \text{if } x < x_0 \\ 2\ln 10 + x + C + a(x_1 - x)^m & x_0 \le x \le x_1 \\ 2\ln 10 + x + C & x_1 < x, \end{cases} \tag{7}$$

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, and we give the corresponding formulae here. The electron density of the medium, n, can be written as

$$n = \begin{cases} N_{Av}\rho\dfrac{Z}{A} & \text{for elements} \\ N_{Av}\rho\dfrac{\sum_i n_i Z_i}{\sum_i n_i A_i} & \text{for compounds/mixtures,} \end{cases} \tag{8}$$

where

N	Avogadro's number;
Z_i	atomic number;
A_i	atomic weight;
ρ	density of the material;
n_i	proportion by number of the i^{th} element in the material (for a mixture $n_i = Wp_i/A_i$ where p_i the proportion by weight and W is the molecular weight).

The average mean ionisation energy can be calculated as [50] [51] [52] [12]:

$$I(GeV) = \begin{cases} 16 \cdot 10^{-9}\ Z^{0.9} & \text{for a chemical element} \\ \exp\left[\sum_i n_i Z_i \ln I(Z_i)/\sum_i n_i Z_i\right] & \text{for a compound or mixture} \end{cases} \tag{9}$$

The density effect correction parameters can be computed (for condensed medium [53]) as

$$\begin{aligned} C &= 1 + 2\ln\frac{I}{28.8 \cdot 10^{-9}\sqrt{\rho}\sum n_i Z_i/\sum n_i A_i} \\ m &= 3 \\ x_a &= \frac{C}{2\ln 10} \\ a &= 2\ln 10\frac{(x_a - x_0)^m}{(x_1 - x_0)} \end{aligned}$$

I	C	x_0	x_1
$< 10^{-7}$	< 3.681	0.2	2
$< 10^{-7}$	≥ 3.681	$-0.326C - 1$	2
$\geq 10^{-7}$	5.215	0.2	3
$\geq 10^{-7}$	≥ 5.215	$-0.326C - 1.5$	3

2.2 Total cross-sections

The integration of formula (2) gives the total cross-section [15], [17] for Möller scattering (e^-e^-):

$$\sigma(Z, E, \mathtt{DCUTE}) = \frac{2\pi r_0^2 m Z}{\beta^2(E - m)}\left[\frac{(\gamma - 1)}{\gamma^2}\left(\frac{1}{x} - 1\right) + \frac{1}{x} - \frac{1}{1 - x} - \frac{2\gamma - 1}{\gamma^2}\ln\frac{1 - x}{x}\right] \tag{10}$$

and for Bhabha scattering (e^+e^-):

$$\sigma(Z, E, \mathtt{DCUTE}) = \frac{2\pi r_0^2 m Z}{(E - m)}\left[\frac{1}{\beta^2}\left(\frac{1}{x} - 1\right) + B_1 x + B_2(1 - x) - \frac{B_3}{2}(1 - x^2) + \frac{B_4}{3}(1 - x^3)\right] \tag{11}$$

where

$$\gamma = \frac{E}{m} \qquad \beta^2 = 1 - \frac{1}{\gamma^2}$$

$$x = \frac{\mathtt{DCUTE}}{E - m} \qquad \gamma = \frac{1}{\gamma + 1}$$

$$B_1 = 2 - y^2 \qquad B_2 = (1 - 2y)(3 + y^2)$$

$$B_3 = (1 - 2y)^2 + (1 - 2y)^3 \qquad B_4 = (1 - 2y)^3$$

The formulae [10] and [11] give the total cross-section of the scattering above the threshold energies

$$T^{thr}_{Moller} = 2\mathtt{DCUTE} \qquad \text{and} \qquad T^{thr}_{Bhabha} = \mathtt{DCUTE} \tag{12}$$

The interaction length for the production of δ-rays is calculated during initialisation by routine GDRSGA.

Origin : D. Ward, L.Urbán
Revision :
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Simulation of the delta-ray production

1 Subroutines

`CALL` **GDRAY**

`GDRAY` generates explicitly the delta-rays (see `[PHYS330]` for treatment of the ionization as continuous energy loss and for the calculation of the total cross-section).

`input:` common `/GCTRAK/`
`output:` common `/GCKING/`

The routine is called from the tracking routines `GTELEC`, `GTMUON`, `GTHION` and `GTHADR` when a charged particle reaches its interaction point.

2 Method

2.1 Differential cross-section

The differential cross-section of the δ-ray production can be written as in equations (1, 2) [54], [12], [55]. For the electron/electron (Möller) scattering we have:

$$\frac{d\sigma}{d\epsilon} = \frac{2\pi Z r_0^2 m}{\beta^2(E-m)}\left[\frac{(\gamma-1)^2}{\gamma^2} + \frac{1}{\epsilon}\left(\frac{1}{\epsilon} - \frac{2\gamma-1}{\gamma^2}\right) + \frac{1}{1-\epsilon}\left(\frac{1}{1-\epsilon}\frac{2\gamma-1}{\gamma^2}\right)\right] \tag{1}$$

and for the positron-electron (Bhabha) scattering:

$$\frac{d\sigma}{d\epsilon} = \frac{2\pi Z r_0^2 m}{(E-m)}\left[\frac{1}{\beta^2\epsilon^2} - \frac{B_1}{\epsilon} + B_2 - B_3\epsilon + B_4\epsilon^2\right] \tag{2}$$

where

Z	$=$	atomic number of the medium	E	$=$	energy of the incident particle
M	$=$	rest mass of the incident particle	γ	$=$	$\frac{E}{M}$
β^2	$=$	$1-\frac{1}{\gamma^2}$	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	$=$	$(1-2y)^3$
ϵ	$=$	$\frac{T}{E-m}$			

with T 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:

$$\epsilon_0 = \frac{\mathtt{TCUT}}{E-m} \le \epsilon \le \frac{1}{2} \quad \text{for } e^-e^- \qquad\qquad \epsilon_0 = \frac{\mathtt{TCUT}}{E-m} \le \epsilon \le 1 \quad \text{for } e^+e^-$$

For the other charged particles the differential cross-section can be written:

$$\begin{aligned}
\frac{d\sigma}{dT} &= 2\pi Z r_0^2 m \frac{1}{\beta^2}\frac{1}{T^2}\left[1-\beta^2\frac{T}{\mathtt{TMAX}}\right] && \text{for spin 0 particle} \\
\frac{d\sigma}{dT} &= 2\pi Z r_0^2 m \frac{1}{\beta^2}\frac{1}{T^2}\left[1-\beta^2\frac{T}{\mathtt{TMAX}}+\frac{T^2}{2E^2}\right] && \text{for spin 1/2 particle}
\end{aligned}$$

where `TMAX` is the maximum energy transferable to the free electron:

$$\mathtt{TMAX} = \frac{2m(\gamma^2-1)}{1+2\gamma(m/M)+(m/M)^2}$$

and `TCUT` is the energy threshold for the δ-ray emission: $\mathtt{TCUT} \le T \le \mathtt{TMAX}$

2.2 Sampling

Apart from the normalisation, the cross-section can be written as

$$\frac{d\sigma}{d\epsilon} = f(\epsilon)g(\epsilon),$$

where, for e^-e^- scattering,

$$\begin{aligned}
f(\epsilon) &= \frac{1}{\epsilon^2}\frac{\epsilon_0}{1-2\epsilon_0} \\
g(\epsilon) &= \frac{4}{9\gamma^2-10\gamma+5}\left[(\gamma-1)^2\epsilon^2-(2\gamma^2+2\gamma-1)\frac{\epsilon}{1-\epsilon}+\frac{\gamma^2}{(1-\epsilon)^2}\right]
\end{aligned}$$

and for e^+e^- scattering

$$\begin{aligned}
f(\epsilon) &= \frac{1}{\epsilon^2}\frac{\epsilon_0}{1-\epsilon_0} \\
g(\epsilon) &= \frac{B_0-B_1\epsilon+B_2\epsilon^2-B_3\epsilon^3+B_4\epsilon^4}{B_0-B_1\epsilon_0+B_2\epsilon_0^2-B_3\epsilon_0^3+B_4\epsilon_0^4}
\end{aligned}$$

Here $B_0 = \gamma^2/(\gamma^2-1)$ and all the other quantities have been defined above. For the other charged particles:

$$\begin{aligned}
f(T) &= \left(\frac{1}{\mathtt{TCUT}}-\frac{1}{\mathtt{TMAX}}\right)\frac{1}{T} \\
g(T) &= 1-\beta^2\frac{T}{\mathtt{TMAX}}+\frac{T^2}{2E^2} \quad \text{(last term for spin-}\tfrac{1}{2}\text{ particle only)}
\end{aligned}$$

`GDRAY` samples the variable ϵ by:

1. sample ϵ from $f(\epsilon)$
2. calculate the rejection function $g(\epsilon)$ and accept 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 the direction of the incident particle. The azimuthal angle ϕ is generated isotropically; the polar angle 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.

Origin : L.Urbán
Revision : L.Urbán
Documentation : F.Carminati, K.Lassila-Perini

Submitted: 10.04.86
Revised: 16.12.93

Simulation of energy loss straggling

1 Subroutines

```
CALL GFLUCT (DEMEAN,DE*)
```

DEMEAN (REAL) average energy loss according to the energy loss tables;
DE (REAL) actual energy loss.

GFLUCT selects the method to sample the fluctuations around the mean energy loss DEMEAN and returns the energy loss DE in the current step.

If δ-rays are not produced (DRAY=0), it calls GLANDG when the current particle and material parameters are in the validity range of Landau theory, GVAVIV when in the range of Vavilov theory, or performs a Gaussian sampling. For fluctuations in small steps GLANDZ is called.

If δ-rays are produced (DRAY=1), it calls GLANDZ which, the cut for the δ-ray production being set, does sampling from restricted formula.

GFLUCT is called from the tracking routines GTELEC, GTHADR and GTMUON when the LOSS flag is set to 1, 2 or 3.

```
CALL GLANDG (YRAN*)
```

YRAN (REAL) random variable distributed according to the Landau distribution;

GLANDG samples from the Landau distribution. It is called from GFLUCT.

```
VALUE = GVAVIV (RKAPPA,BETA2,RAN)
```

RKAPPA (REAL) κ parameter of the Vavilov distribution (see below);
BETA2 (REAL) β^2 of the particle;
RAN (REAL) random number uniformly distributed in]0,1[.

GVAVIV samples samples the variable $\lambda = \lambda_v/\kappa - \ln\kappa$, sometimes called the Landau λ, where λ_v is distributed according to the Vavilov distribution. For more details see below. It is called from GFLUCT.

```
CALL GLANDZ (Z,STEP,P,E,DEDX,DE*,POTI,POTIL)
```

Z (REAL) atomic number of the medium;
STEP (REAL) step size in cm;
P (REAL) momentum of the particle in GeV c^{-1};
E (REAL) total energy of the particle in GeV;
DEDX (REAL) average energy loss in the step according to the energy loss tables;
DE (REAL) actual energy loss in the step;
POTI (REAL) average ionisation potential for the medium;
POTIL (REAL) logarithm of the average ionisation potential.

GLANDZ gives the energy loss DE for the current particle with energy E and momentum P in a material of atomic number Z where the average energy loss is DEDX and the current step length STEP. It is called from GFLUCT for restricted Landau fluctuations when δ-rays are produced or to sample the fluctuations in thin layers (with or without δ-ray production).

2 Method

Due to the statistical nature of ionisation energy loss, large fluctuations can occur in the amount of energy deposited by a particle traversing an absorber element. As recently reviewed [56], continuous processes such as multiple scattering and energy loss play a relevant role in the longitudinal and lateral development of electromagnetic and hadronic showers, and in the case of sampling calorimeters the measured resolution can be significantly affected 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

$$\kappa = \frac{\xi}{E_{max}}$$

E_{max} is the maximum transferable energy in a single collision with an atomic electron.

$$E_{max} = \frac{2m_e\beta^2\gamma^2}{1 + 2\gamma m_e/m_x + (m_e/m_x)^2},$$

where $\gamma = E/m_x$, E is energy and m_x the mass of the incident particle, $\beta^2 = 1 - 1/\gamma^2$ and m_e is the electron mass. ξ comes from the Rutherford scattering cross section and is defined as:

$$\xi = \frac{2\pi z^2 e^4 N_{Av} Z \rho \delta x}{m_e \beta^2 c^2 A} = 153.4 \frac{z^2}{\beta^2} \frac{Z}{A} \rho \delta x \quad \text{keV},$$

where

z	charge of the incident particle
N_{Av}	Avogadro's number
Z	atomic number of the material
A	atomic weight of the material
ρ	density
δx	thickness of the material

κ measures the contribution of the collisions with energy transfer close to E_{max}. For a given absorber, κ tends towards large values if δx is large and/or if β is small. Likewise, κ tends towards zero if δx is small and/or if β approaches 1.

The value of κ distinguishes two regimes which occur in the description of ionisation fluctuations :

1. 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 $\kappa > 10$ (i.e. when the mean energy loss in the absorber is greater than the maximum energy transfer in a single collision).

2. Particles traversing thin counters and incident electrons under any conditions.

   The relevant inequalities and distributions are $0.01 < \kappa < 10$, Vavilov distribution, and $\kappa < 0.01$, Landau distribution.

An additional regime is defined by the contribution of the collisions with low energy transfer which can be estimated with the relation ξ/I_0, where I_0 is the mean ionisation potential of the atom. Landau theory assumes that the number of these collisions is high, and consequently, it has a restriction $\xi/I_0 \gg 1$. In `GEANT`, the limit of Landau theory has been set at $\xi/I_0 = 50$. Below this limit special models taking into account the atomic structure of the material are used. This is important in thin layers and gaseous materials. Figure 37 shows the behaviour of ξ/I_0 as a function of the layer thickness for an electron of 100 keV and 1 GeV of kinetic energy in Argon, Silicon and Uranium.

In the following sections, the different theories and models for the energy loss fluctuation are described. First, the Landau theory and its limitations are discussed, and then, the Vavilov and Gaussian straggling functions and the methods in the thin layers and gaseous materials are presented.

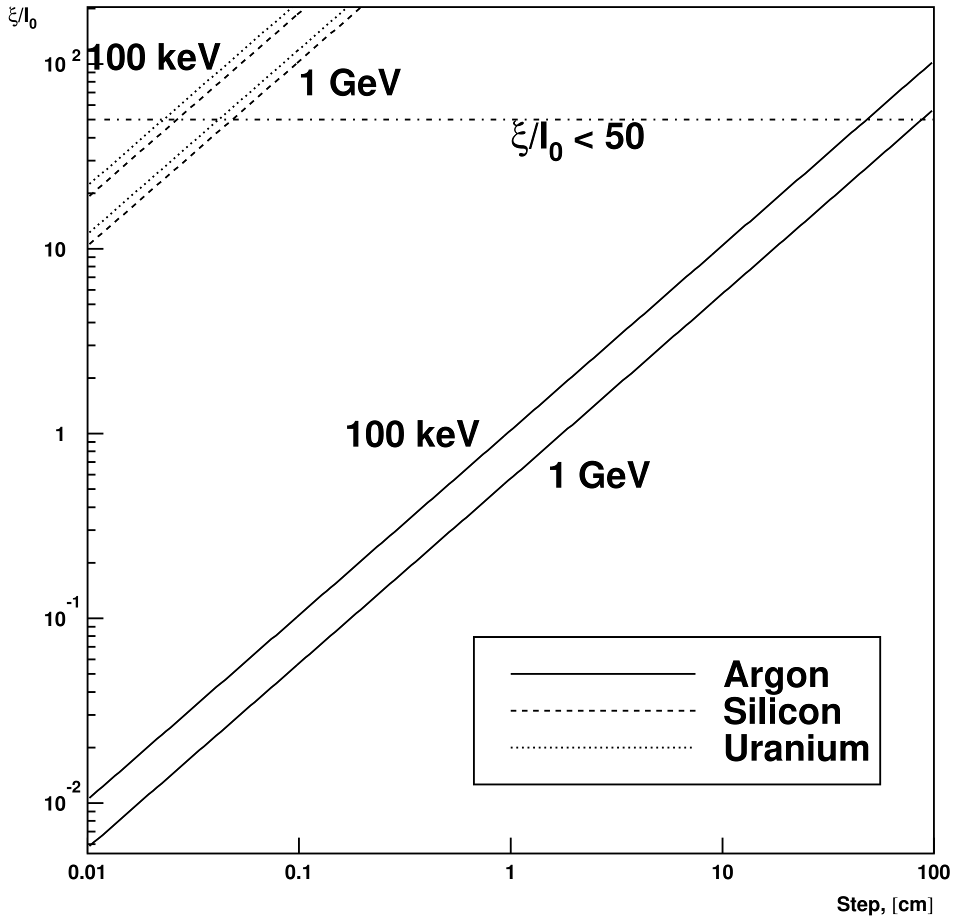


Figure 37: The variable ξ/I_0 can be used to measure the validity range of the Landau theory. It depends on the type and energy of the particle, Z, A and the ionisation potential of the material and the layer thickness.

2.1 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 [57], [58] function $\phi(\lambda)$ as:

$$f(\epsilon, \delta x) = \frac{1}{\xi}\phi(\lambda)$$

where

$$\begin{aligned}
\phi(\lambda) &= \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} \exp\left(u \ln u + \lambda u\right) du \qquad\qquad c \geq 0 \\
\lambda &= \frac{\epsilon - \bar{\epsilon}}{\xi} - \gamma' - \beta^2 - \ln\frac{\xi}{E_{max}} \\
\gamma' &= 0.422784\ldots = 1 - \gamma \\
\gamma &= 0.577215\ldots \text{(Euler's constant)} \\
\bar{\epsilon} &= \text{average energy loss} \\
\epsilon &= \text{actual energy loss}
\end{aligned}$$

Restrictions

The Landau formalism makes two restrictive assumptions :

1. The typical energy loss is small compared to the maximum energy loss in a single collision. This restriction is removed in the Vavilov theory (see section 2.2).

2. The typical energy loss in the absorber should be large compared to the binding energy of the most tightly bound electron. For gaseous detectors, typical energy losses are a 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 [59]) is necessary to accurately simulate data distributions. In `GEANT`, a parameterised model by L. Urbán is used (see section 2.4).

In addition, the average value of the Landau distribution is infinite. Summing the Landau fluctuation obtained to the average energy from the dE/dx tables, we obtain a value which is larger than the one coming from the table. The probability to sample a large value is small, so it takes a large number of steps (extractions) for the average fluctuation to be significantly larger than zero. This introduces a dependence of the energy loss on the step size which can affect calculations.

A solution to this has been to introduce a limit on the value of the variable sampled by the Landau distribution [60], in order to keep the average fluctuation to 0. The value obtained from the `GLANDO` routine is:

$$\delta dE/dx = \epsilon - \bar{\epsilon} = \xi(\lambda - \gamma' + \beta^2 + \ln \frac{\xi}{E_{max}})$$

In order for this to have average 0, we must impose that:

$$\bar{\lambda} = -\gamma' - \beta^2 - \ln \frac{\xi}{E_{max}}$$

This is realised introducing a $\lambda_{max}(\bar{\lambda})$ such that if only values of $\lambda \leq \lambda_{max}$ are accepted, the average value of the distribution is $\bar{\lambda}$.

A parametric fit to the universal Landau distribution has been performed, with following result:

$$\lambda_{max} = 0.60715 + 1.1934\bar{\lambda} + (0.67794 + 0.052382\bar{\lambda})\exp(0.94753 + 0.74442\bar{\lambda})$$

only values smaller than λ_{max} are accepted, otherwise the distribution is resampled.

2.2 Vavilov theory

Vavilov [61] derived a more accurate straggling distribution by introducing the kinematic limit on the maximum transferable energy in a single collision, rather than using $E_{max} = \infty$. Using the notations of [62] we can write:

$$f(\epsilon, \delta s) = \frac{1}{\xi}\phi_v\left(\lambda_v, \kappa, \beta^2\right)$$

where

$$\begin{aligned}
\phi_v\left(\lambda_v, \kappa, \beta^2\right) &= \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} \phi(s)\, e^{\lambda s} ds \qquad\qquad c \geq 0 \\
\phi(s) &= \exp\left[\kappa(1+\beta^2\gamma)\right]\ \exp\left[\psi(s)\right], \\
\psi(s) &= s\ln\kappa + (s+\beta^2\kappa)\left[\ln(s/\kappa) + E_1(s/\kappa)\right] - \kappa e^{-s/\kappa},
\end{aligned}$$

and

$$
\begin{aligned}
E_1(z) &= \int_z^\infty t^{-1}e^{-t}dt \qquad \text{(the exponential integral)} \\
\lambda_v &= \kappa\left[\frac{\epsilon-\bar{\epsilon}}{\xi}-\gamma'-\beta^2\right]
\end{aligned}
$$

The Vavilov parameters are simply related to the Landau parameter by $\lambda_L = \lambda_v/\kappa - \ln\kappa$. It can be shown that as $\kappa \to 0$, the distribution of the variable λ_L approaches that of Landau. For $\kappa \leq 0.01$ the two distributions are already practically identical. Contrary to what many textbooks report, the Vavilov distribution *does not* approximate the Landau distribution for small κ, but rather the distribution of λ_L defined above tends to the distribution of the true λ from the Landau density function. Thus the routine GVAVIV samples the variable λ_L rather than λ_v. For $\kappa \geq 10$ the Vavilov distribution tends to a Gaussian distribution (see next section).

2.3 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 [62] it has been demonstrated by Seltzer and Berger [63] that for $\kappa \geq 10$ the Vavilov distribution can be replaced by a Gaussian of the form :

$$
f(\epsilon,\delta s) \approx \frac{1}{\xi\sqrt{\frac{2\pi}{\kappa}(1-\beta^2/2)}}\exp\left[\frac{(\epsilon-\bar{\epsilon})^2}{2}\frac{\kappa}{\xi^2(1-\beta^2/2)}\right]
$$

thus implying

$$
\begin{aligned}
\text{mean} &= \bar{\epsilon} \\
\sigma^2 &= \frac{\xi^2}{\kappa}(1-\beta^2/2) = \xi E_{max}(1-\beta^2/2)
\end{aligned}
$$

2.4 Urbán model

The method for computing restricted energy losses with δ-ray production above given threshold energy in GEANT is a Monte Carlo method that can be used for thin layers. It is fast and it can be used for any thickness of a medium. Approaching the limit of the validity of Landau's theory, the loss distribution approaches smoothly the Landau form as shown in figure 38.

It is assumed that the atoms have only two energy levels with binding energy E_1 and E_2. The particle–atom interaction will then be an excitation with energy loss E_1 or E_2, or an ionisation with an energy loss distributed according to a function $g(E) \sim 1/E^2$:

$$
g(E) = \frac{(E_{max}+I)I}{E_{max}}\frac{1}{E^2} \tag{1}
$$

The macroscopic cross-section for excitations ($i = 1, 2$) is

$$
\Sigma_i = C\frac{f_i}{E_i}\frac{\ln(2m\beta^2\gamma^2/E_i)-\beta^2}{\ln(2m\beta^2\gamma^2/I)-\beta^2}(1-r) \tag{2}
$$

and the macroscopic cross-section for ionisation is

$$
\Sigma_3 = C\frac{E_{max}}{I(E_{max}+I)\ln(\frac{E_{max}+I}{I})}r \tag{3}
$$

E_{max} is the GEANT cut for δ-production, or the maximum energy transfer minus mean ionisation energy, if it is smaller than this cut-off value. The following notation is used:

r, C	parameters of the model
E_i	atomic energy levels
I	mean ionisation energy
f_i	oscillator strengths

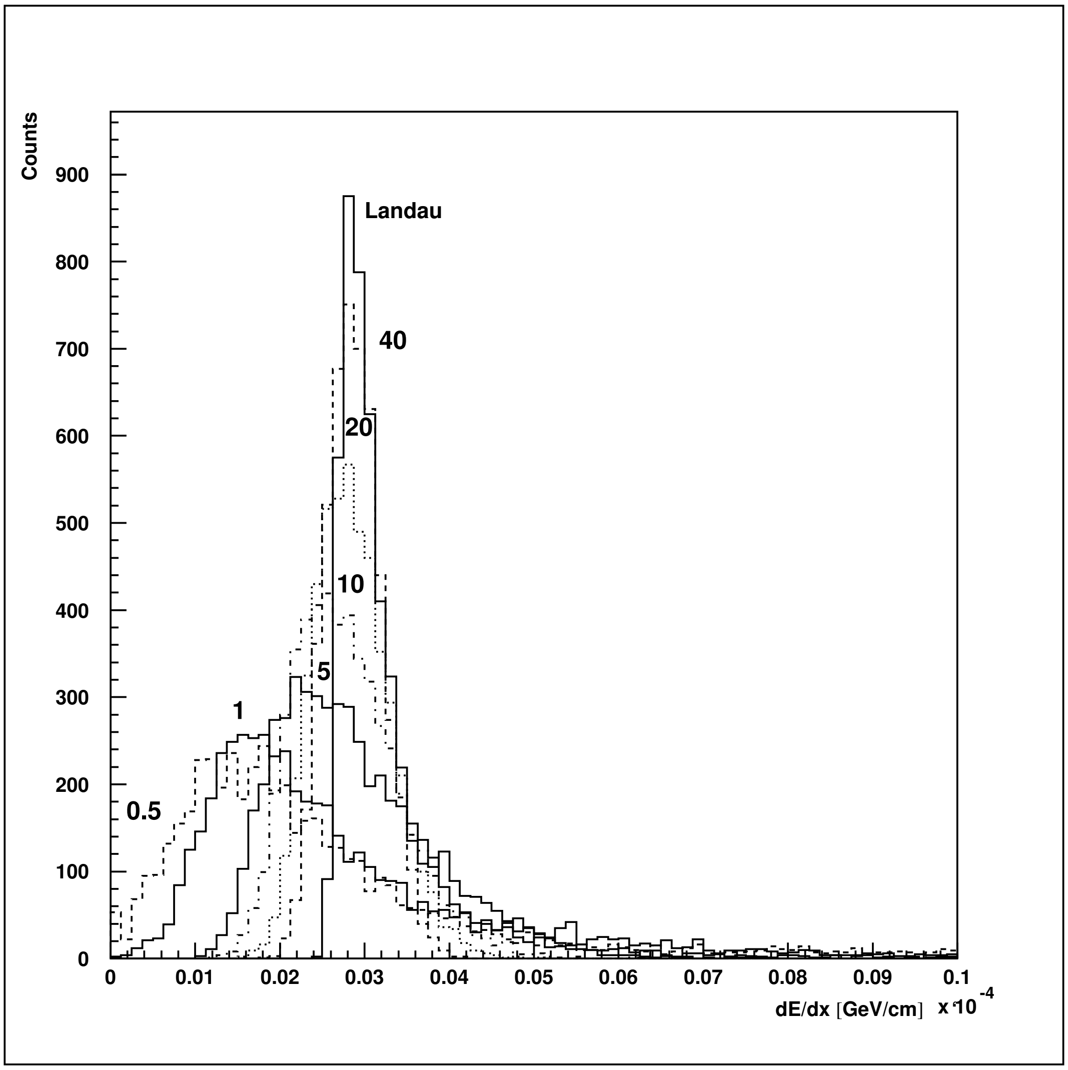


Figure 38: Energy loss distribution for a 3 GeV electron in Argon as given by standard GEANT. The width of the layers is given in centimeters.

The model has the parameters f_i, E_i, C and r $(0 \leq r \leq 1)$. The oscillator strengths f_i and the atomic level energies E_i should satisfy the constraints

$$f_1 + f_2 = 1 \tag{4}$$

$$f_1 \ln E_1 + f_2 \ln E_2 = \ln I \tag{5}$$

The parameter C can be defined with the help of the mean energy loss dE/dx in the following way: The numbers of collisions (n_i, i = 1,2 for the excitation and 3 for the ionisation) follow the Poissonian distribution with a mean number $\langle n_i \rangle$. In a step Δx the mean number of collisions is

$$\langle n_i \rangle = \Sigma_i \Delta x \tag{6}$$

The mean energy loss dE/dx in a step is the sum of the excitation and ionisation contributions

$$\frac{dE}{dx}\Delta x = \left[\Sigma_1 E_1 + \Sigma_2 E_2 + \Sigma_3 \int_I^{E_{max}+I} E\, g(E)\, dE\right]\Delta x \tag{7}$$

From this, using the equations (2), (3), (4) and (5), one can define the parameter C

$$C = \frac{dE}{dx} \tag{8}$$

The following values have been chosen in GEANT for the other parameters:

$$f_2 = \begin{cases} 0 & \text{if } Z \leq 2 \\ 2/Z & \text{if } Z > 2 \end{cases} \quad \Rightarrow \quad f_1 = 1 - f_2$$

$$E_2 = 10Z^2 \text{ eV} \quad \Rightarrow \quad E_1 = \left(\frac{I}{E_2^{f_2}}\right)^{\frac{1}{f_1}}$$

$$r = 0.4$$

With these values the atomic level E_2 corresponds approximately the K-shell energy of the atoms and Zf_2 the number of K-shell electrons. r is the only variable which can be tuned freely. It determines the relative contribution of ionisation and excitation to the energy loss.

The energy loss is computed with the assumption that the step length (or the relative energy loss) is small, and — in consequence — the cross-section can be considered constant along the path length. The energy loss due to the excitation is

$$\Delta E_e = n_1 E_1 + n_2 E_2 \tag{9}$$

where n_1 and n_2 are sampled from Poissonian distribution as discussed above. The loss due to the ionisation can be generated from the distribution $g(E)$ by the inverse transformation method:

$$\begin{aligned} u = F(E) &= \int_I^E g(x)dx \\ E = F^{-1}(u) &= \frac{I}{1 - u\frac{E_{max}}{E_{max}+I}} \end{aligned} \tag{10}$$

(11)

where u is a uniform random number between $F(I) = 0$ and $F(E_{max} + I) = 1$. The contribution from the ionisations will be

$$\Delta E_i = \sum_{j=1}^{n_3} \frac{I}{1 - u_j \frac{E_{max}}{E_{max}+I}} \tag{12}$$

where n_3 is the number of ionisation (sampled from Poissonian distribution). The energy loss in a step will then be $\Delta E = \Delta E_e + \Delta E_i$.

Fast simulation for $n_3 \geq 16$

If the number of ionisation n_3 is bigger than 16, a faster sampling method can be used. The possible energy loss interval is divided in two parts: one in which the number of collisions is large and the sampling can be done from a Gaussian distribution and the other in which the energy loss is sampled for each collision. Let us call the former interval $[I, \alpha I]$ the interval A, and the latter $[\alpha I, E_{max}]$ the interval B. α lies between 1 and E_{max}/I. A collision with a loss in the interval A happens with the probability

$$P(\alpha) = \int_I^{\alpha I} g(E)\, dE = \frac{(E_{max} + I)(\alpha - 1)}{E_{max}\alpha} \tag{13}$$

The mean energy loss and the standard deviation for this type of collision are

$$\langle \Delta E(\alpha) \rangle = \frac{1}{P(\alpha)} \int_I^{\alpha I} E\, g(E)\, dE = \frac{I\alpha \ln \alpha}{\alpha - 1} \tag{14}$$

and

$$\sigma^2(\alpha) = \frac{1}{P(\alpha)} \int_I^{\alpha I} E^2\, g(E)\, dE = I^2\alpha \left(1 - \frac{\alpha \ln^2\alpha}{(\alpha - 1)^2}\right) \tag{15}$$

If the collision number is high , we assume that the number of the type A collisions can be calculated from a Gaussian distribution with the following mean value and standard deviation:

$$\langle n_A \rangle = n_3 P(\alpha) \tag{16}$$
$$\sigma_A^2 = n_3 P(\alpha)(1 - P(\alpha)) \tag{17}$$

It is further assumed that the energy loss in these collisions has a Gaussian distribution with

$$\langle \Delta E_A \rangle = n_A \langle \Delta E(\alpha) \rangle \tag{18}$$
$$\sigma_{E,A}^2 = n_A \sigma^2(\alpha) \tag{19}$$

The energy loss of these collision can then be sampled from the Gaussian distribution.
The collisions where the energy loss is in the interval B are sampled directly from

$$\Delta E_B = \sum_{i=1}^{n_3 - n_A} \frac{\alpha I}{1 - u_i \frac{E_{max} + I - \alpha I}{E_{max} + I}} \tag{20}$$

The total energy loss is the sum of these two types of collisions:

$$\Delta E = \Delta E_A + \Delta E_B \tag{21}$$

The approximation of equations ((16), (17), (18) and (19) can be used under the following conditions:

$$\langle n_A \rangle - c\, \sigma_A \geq 0 \tag{22}$$
$$\langle n_A \rangle + c\, \sigma_A \leq n_3 \tag{23}$$
$$\langle \Delta E_A \rangle - c\, \sigma_{E,A} \geq 0 \tag{24}$$

where $c \geq 4$. From the equations (13), (16) and (18) and from the conditions (22) and (23) the following limits can be derived:

$$\alpha_{min} = \frac{(n_3 + c^2)(E_{max} + I)}{n_3(E_{max} + I) + c^2 I} \leq \alpha \leq \alpha_{max} = \frac{(n_3 + c^2)(E_{max} + I)}{c^2(E_{max} + I) + n_3 I} \tag{25}$$

This conditions gives a lower limit to number of the ionisations n_3 for which the fast sampling can be done:

$$n_3 \geq c^2 \tag{26}$$

As in the conditions (22), (23) and (24) the value of c is as minimum 4, one gets $n_3 \geq 16$. In order to speed the simulation, the maximum value is used for α.
The number of collisions with energy loss in the interval B (the number of interactions which has to be simulated directly) increases slowly with the total number of collisions n_3. The maximum number of these collisions can be estimated as

$$n_{B,max} = n_3 - n_{A,min} \approx n_3(\langle n_A \rangle - \sigma_A) \tag{27}$$

From the previous expressions for $\langle n_A \rangle$ and σ_A one can derive the condition

$$n_B \leq n_{B,max} = \frac{2 n_3 c^2}{n_3 + c^2} \tag{28}$$

The following values are obtained with $c = 4$:

n_3	$n_{B,max}$	n_3	$n_{B,max}$
16	16	200	29.63
20	17.78	500	31.01
50	24.24	1000	31.50
100	27.59	∞	32.00

Special sampling for lower part of the spectrum

If the step length is very small ($\leq$ 5 mm in gases, $\leq$ 2-3 μm in solids) the model gives 0 energy loss for some events. To avoid this, the probability of 0 energy loss is computed

$$P(\Delta E = 0) = e^{-(\langle n_1 \rangle + \langle n_2 \rangle + \langle n_3 \rangle)} \tag{29}$$

If the probability is bigger than 0.01 a special sampling is done, taking into account the fact that in these cases the projectile interacts only with the outer electrons of the atom. An energy level $E_0 = 10$ eV is chosen to correspond to the outer electrons. The mean number of collisions can be calculated from

$$\langle n \rangle = \frac{1}{E_0} \frac{dE}{dx} \Delta x \tag{30}$$

The number of collisions n is sampled from Poissonian distribution. In the case of the thin layers, all the collisions are considered as ionisations and the energy loss is computed as

$$\Delta E = \sum_{i=1}^{n} \frac{E_0}{1 - \frac{E_{max}}{E_{max}+E_0} u_i} \tag{31}$$

3 Implementation

The method to be used for energy loss straggling is chosen in `GFLUCT`. If δ-rays are produced (`DRAY = 1`) above the cut value `DCUTE` and the detailed PAI simulation for straggling in thin layers (see PHYS334) is not chosen (`STRA = 0`, default), `GLANDZ` is called always as it samples from the restricted distribution (the energy loss of the δ-rays which are explicitly produced should not be taken into the energy loss distribution).

If δ-rays are not produced, the values of κ and ξ/I are computed, and Urbán, Landau, Vavilov or Gaussian model is chosen accordingly. If Urbán model is used, `GLANDZ` will be called with infinite (`BIG`) value for the δ-ray production.

When Landau model is chosen, it is possible to use a routine from the CERN Program Library to sample random numbers from the Landau distribution : `GENLAN` written by James and Hancock [64] and copied into `GEANT` source file as `GLANDG`. This routine has been modified in order to reproduce the tail for large values of X. The original routine did not give values of X larger than 200.

```
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, a *simpler form* of $\phi(\lambda)$ had been proposed by Moyal [65] and used on occasions which has little in common with the function defined in section 2.1 [66].

For the Vavilov distribution we have used the function `GVAVIV` by Rotondi and Montagna [67].

Origin : M.Maire
Revision : F.Carminati
Documentation :

Submitted: 30.05.86
Revised: 16.12.93

Information about energy loss fluctuations

1 Subroutines

```
CALL GDRPRT (IPART,IMATE,STEP,NPOINT)
```

IPART (INTEGER) GEANT particle number;
IMATE (INTEGER) GEANT material number;
STEP (REAL) step in cm;
NPOINT (INTEGER) number of logarithmically spaced energy points between EKMIN (default 10 keV) and EKMAX (default 10 TeV). See [BASE040] for more information on these value and how to change them with the ERAN input record;

This routine calculates and prints the relevant parameters relative to the simulation of energy loss fluctuations due to ionisation and δ-ray production in STEP cm:

N d-rays average number of δ-rays produced;
dE/I average energy loss divided by the average ionisation potential;
xi/I variable ξ/I of the Landau/Vavilov theory;
xi/Emax variable ξ/E_{Max} of the Landau/Vavilov theory;
regime valid model to simulate energy loss fluctuations. Possible values are:

- Gauss — Gauss distribution;
- Vavilov — Vavilov distribution;
- Landau — Landau distribution;
- Urbán — GLANDZ routine of Urban (see [PHYS332]);
- ASHO — atomic harmonic oscillator approximation (see [PHYS334]);
- PAI — Photo-Absorption Ionisation model (see [PHYS334]);

This routine has no action on the GEANT system. It is intended as a help to users understanding the model which GEANT may be asked to use, for instance via the routine GSTPAR [CONS210].

2 Method

When a charged particle traverses a portion of matter, it interacts with the electrons and the nuclei of the atoms. Most of these interactions are electromagnetic (quasi-) elastic collisions in which the incoming particle loses energy in the laboratory reference frame. The amount of energy loss in a thickness of material t is subject to two sources of fluctuations. The *number* of collisions can fluctuate, and at the same time the *energy* lost in each collision varies statistically. Both distributions are characterised by a Poissonian-like behaviour. We can distinguish between collisions where the energy transferred to the atomic electrons is enough to extract them from the atoms (ionisation with production of δ-rays) and collisions where the atomic structure is excited, without a complete ionisation (excitation). It has to be noted that the energy transferred to the nuclei is usually negligible. Momentum conservation considerations show that the ratio of the energy transferred to electrons to the energy transferred to nucleus in Coulomb interactions is of the order of $m_e Z A^{-1} m_p^{-1}$.

Several theories have been proposed to describe this important mechanism in the transport of charged particles. The main difference of these theories is in the greater or lesser detail with which the scattering centres are described. The detail of these theories as they are implemented in GEANT can be found in [PHYS332], [PHYS334] for energy loss fluctuations and in [PHYS430], [PHYS431] for energy loss.

In general, the greater the thickness of the layer traversed in terms of the number of atoms encountered, the larger is the number of collisions. In this case a detailed description of the atomic structure may be irrelevant to account for the form of the fluctuation. Landau and Vavilov have proposed theories in this region and these are implemented in GEANT.

When the thickness of the material is such that the number of collisions becomes smaller, the detailed nature of the atomic structure becomes important in determining the fluctuations of the energy loss. In this case the coupling of the various atomic energy levels to the to the Coulomb field must be taken into account.

When simulating the energy loss by ionisation, the average path-length between collisions is a function of the cross section $\sigma(E)$. This cross section becomes very large when $E \rightarrow 0$, so that it is necessary to establish a threshold energy below which the process is described in a *condensed* way. Above this threshold, ionisations are described in a detailed way, with the production of δ-rays. Thus, when a charged particle is moving in a medium, there are in general two ways to simulate the energy loss by ionisation:

1. calculate the average value the energy loss via the full Bethe-Bloch equation. This takes into account δ-rays generation. The fluctuations are then explicitly introduced via an appropriate distribution. In GEANT this method is selected by the value `ILOSS = 2` which is controlled by the LOSS data record.

2. explicitly generate the δ-rays *above* a given threshold (DCUTE for electrons, DCUTM for others particle). In this case the average energy loss is computed from a restricted formula and both the value of the energy loss and the number of δ-rays produced are function of the threshold cut (hereafter DRCUT). In GEANT this method is selected by the value `ILOSS = 1` which is controlled by the LOSS data record.

2.1 Validity ranges for the different models

The limits of the validity for the various models are somewhat arbitrary within a certain range of material densities and particle energies. To help determine these limits, some characteristic parameters are used:

ξ typical energy loss of a particle in a given layer of material. Its value is:

$$\xi = 2\pi r_e^2 m_e c^2 N_{Av} \left(\frac{Z_{inc}}{\beta}\right)^2 \frac{Z\rho t}{A} = 0.1535 \left(\frac{Z_{inc}}{\beta}\right)^2 \frac{Z\rho t}{A}\,(MeV)$$

E_{Max} maximum transferable energy in a single collision:

$$E_{Max} = \frac{2m_e\beta^2\gamma^2}{1 + 2\gamma\frac{m_e}{M} + \left(\frac{m_e}{M}\right)^2}$$

where M is the mass of the incoming particle. Note that for incoming electrons this value has to be divided by 2 due to the impossibility distinguishing the two electrons in the final state.

I typical electron binding energy. This is a value which characterises the average energy levels of the atomic electrons. In GEANT it is parametrised as:

$$I = 16Z^{0.9}\,(eV)$$

$\frac{dE}{dx}\,t$ average energy loss by the particle in a layer of thickness t according to the GEANT energy loss tables.

ξ/E_{Max} (hereafter κ) relative importance of high energy transfer collisions in the ionisation process

$\min\left(\frac{dE}{dx}\,t, \xi\right)/I$ (hereafter Δ) estimation of the number of collisions with energy close to the ionisation energy.

In GEANT two variables control the model used to describe the energy loss fluctuations:

ILOSS fluctuation model:

	`0`	no energy loss;
	`1`	δ-rays are produced above the threshold, *reduced* fluctuations from δ-rays below the threshold are added to the energy loss;
	`2`	no δ-rays are produced, complete fluctuations are calculated;
	`3`	same as `1`;
	`4`	no fluctuations;
`ISTRA`	energy loss model for *thin* layer (see below):	
	`0`	Urbán model;
	`1`	PAI model;
	`2`	ASHO model for $1 < \Delta \leq 30$,(not yet available) PAI model otherwise;

The validity limits of the different models are estimated as follows:

1. **large number of low-energy collisions:** $\Delta \geq 50$

   if δ-rays generation is requested (`ILOSS = 1` or `3` and `IDRAY = 1`) the Urbán model is used. If δ-rays are not produced (`ILOSS = 2` and `IDRAY = 0`) we distinguish three regions:

   (a) **very few energy transfers close to the maximum:** $\kappa \leq 10^{-2}$
   the Landau distribution is used;

   (b) **few energy transfers close to the maximum:** $10^{-2} < \kappa \leq 10$
   the Vavilov distribution is used;

   (c) **many energy transfers close to the maximum:** $\kappa > 10$
   the Gauss distribution is used;

2. **small number of low-energy collisions:** $\Delta < 50$

   in this region the same model is used for $1 \leq$ `ILOSS` ≤ 3 irrespective of the value of `IDRAY`. The model used depends on the value of `ISTRA`.

   (a) $30 < \Delta \leq 50$
   Urbán model if `ISTRA` $= 0$, PAI model otherwise.

   (b) $1 < \Delta \leq 30$
   Urbán model if `ISTRA` $= 0$, PAI model if `ISTRA` $= 1$ and ASHO model if `ISTRA` $= 2$.

   (c) $0.01 < \Delta \leq 1$
   Urbán model if `ISTRA` $= 0$, PAI model otherwise.

   (d) $\Delta \leq 0.01$
   Urbán model if `ISTRA` $= 0$, PAI model otherwise.

2.2 Recommendations

The full energy loss fluctuations or the restricted fluctuations complemented by the production of δ-rays give equivalent results if the number of δ-rays generated is sufficient, e.g. a few tens along the full trajectory of the particle in the medium.

For a relativistic particle, the number of delta-rays produced per cm can be estimated integrating the cross section provided in [55]:

$$\frac{dN}{dx} \approx \frac{D}{2}\frac{\rho Z}{A}\frac{Z_{inc}^2}{\mathtt{DRCUT}} \tag{1}$$

where

D	0.307 MeV cm^2 g^{-1};
ρ	density of the medium;

Z, A	atomic number and atomic weight of the medium;
Z_{inc}	charge of the particle;
DRCUT	energy threshold for emitted delta-rays.

This formula holds for electrons/positrons as well as for other particles. The number of δ-rays must be sufficient to reproduce the statistical fluctuations in the energy loss, but not too large, as in this case a large amount of time could be spent in tracking them without a corresponding improvement of the energy loss distribution.

On the other hand, the distribution of the energy *lost* by a particle does not necessarily correspond to the distribution of the energy *deposited* by the same particle in the medium traversed. This is particularly true for light or *thin* materials where δ-rays can escape into the next material.

It is the responsibility of the user to estimate the number of δ-rays per cm that are needed, according to the considerations above. Then the correct value for DRCUT should be set with the help of GDRPRT. It is recommended to use the same value for DCUTE and DCUTM. As the GEANT tables for cross-sections are not generated below EKMIN, DRCUT cannot be smaller than EKMIN. The value of EKMIN can be changed via the ERAN data record, and its default value is 10 keV.

In a *light* material like gas, the number of emitted δ-rays for DRCUT $= 10$ keV may not be sufficient to ensure a correct energy loss distribution. If the user is mainly interested in the energy lost by the particle, no discrete δ-ray should be produced setting ILOSS $= 2$ (see [PHYS332]).

In case the user is more interested in a precise simulation of the energy deposited, then the explicit δ-ray generation should be tried.

As far as *thin* materials are concerned ($\Delta < 50$), the default in GEANT is to use the Urbán model. Comparison with experimental data have shown that this model gives very good results, and it is considerably faster than both the ASHO and the PAI models. Users should understand well their setup and their results before they try with different models, which, in principle, should give very similar results.

2.3 Default values in GEANT

In order to avoid double counting, there is an automatic protection in the code: *if ILOSS=2, IDRAY is set to 0 and the generation of δ-rays is disabled.*

The following table summarises the different cases:

	LOSS=2 (default)	LOSS=1
IDRAY	0	1
DCUTE	10^4 GeV	CUTELE
DCUTM	10^4 GeV	CUTELE

Where LOSS$= 1$ means that the restricted fluctuation is activated and LOSS$= 2$ means that the complete Landau/Vavilov/Gauss fluctuations are used in the region $\Delta \geq 50$.

Origin : I.Gavrilenko, P.Nevski, K.Lassila-Perini
Revision :
Documentation : K.Lassila-Perini

Submitted: 11.03.94
Revised: 11.03.94

Models for energy loss fluctuations in thin layers

1 Subroutines

```
CALL GSTINI
```

`GSTINI` initializes the tables that are used for sampling of the energy loss, if the user has set the flag `STRA = 1`. It calls the subroutine `GSTXIN` to integrate an expression for the number of collisions and `GSTTAB` for preparing the tables for different Lorentz factors. It is called from `GPHYSI` at initialization time.

```
CALL GSTXIN
```

`GSTXIN` computes the values needed for the sampling tables. It uses the functions `GOSCIN` to integrate over the photoelectric cross-sections and the function `GKOKRI` to compute the real part of the complex dielectric constant. `GSTXIN` is called from `GSTINI`.

```
VALUE = GOSCIN (EIN1EV,EIN2EV)
```

`GOSCIN` integrates the parameterization of the photoelectric cross-sections from energy `EIN1EV` to energy `EIN2EV`. It is called from `GSTXIN`.

```
VALUE = GKOKRI (E,EMINEV,EMAXEV)
```

`GKOKRI` computes the real part of the complex dieletric constant which is needed in the preparation of energy loss sampling tables. `E` in the energy for which the values are being tabulated, `EMINEV` and `EMAXEV` are the integration limits. `GKOKRI` is called from `GSTXIN`.

```
CALL GSTTAB (GAM,NT,EN,FN)
```

`GSTTAB` prepares the energy loss sampling tables for different Lorentz factors. The input `GAM` is the Lorentz factor, the output values are: `NT` the lenght of the table, `EN` the energy array and `FN` the cumulative probabilty array. It calls `GXGINT` to integrate over the probability function `GSTDN` to get the value of `FN`. `GSTTAB` is called from `GSTINI`.

```
VALUE = GXGINT (EXT,A,B,EPS)
```

`GXGINT` performs the Gaussian integration from `A` to `B` over the function `EXT` with accuracy `EPS`. It is called from `GSTTAB` to integrate over the probability function `GSTDN`.

```
VALUE = GSTDN (LGE)
```

`GSTDN` is the function which gives the probability for a collision with energy transfer `LGE`. It is integrated to get the cumulative probabilty function to be used in the sampling of the energy loss.

```
CALL GSTREN (GAMMA,ECUT,STEP)
```

`GSTREN` computes the energy loss for a particle of Lorentz factor `GAMMA` in a step of length `STEP` if the user has set the flag `STRA = 1`. `ECUT` is the cut for the δ-ray production. `GSTREN` is called from `GFLUCT`.

2 The method

In thin layers, the Landau model of energy loss fluctuations is not valid, because the number of collisions is too small. In this case, the atomic structure of the atom has to be taken into account. The photoabsorption ionization (PAI) model uses the photoelectric cross-sections to describe the energy loss distribution. The results (the width and the most probable value of the energy loss distribution function) given by this model are equal to those given by standard `GEANT` procedure described in `[PHYS332]`. In addition, however, it gives an estimate of the number of collision in each step (`NICOLL` in the common block `GCSTRA`). PAI model is slower than the standard `GEANT` method.

PAI model is activated, if the user set the flag `STRA = 1`. The default value is 0.

2.1 The photoabsorption ionization model

An expression for the distribution of energy loss can be derived considering the energy loss as the sum of the energy transfers in the electromagnetic interactions between the particle and the atom. As the interaction is small (i.e. the energy transfer is small compared to the energy of the passing particle), Born approximation can be used in the perturbation theory. In the derivation, the atomic transition current is considered as a sum of the transition currents of its electrons.

ε is the complex dielectric constant of a medium which describes the electromagnetic properties of the medium and thus the effect of the field of an atom on the energy loss of the particle.

The complex dielectric constant can be written $\varepsilon = \varepsilon_1 + i\varepsilon_2$ where ε_1 describes the polarization and ε_2 the absorptive properties of the medium. ε_2 can be expressed with the help of the oscillator strength function $f(k,\omega)$ which describes the coupling of the electrons to the field of the atom.

$$\varepsilon_2 = \frac{2\pi^2 N e^2}{m\omega} f(k,\omega) \tag{1}$$

m is the mass of the electron and N is the electron density. In a simplified model, the photoabsorption cross-section $\sigma_\gamma(\omega)$ can be used for description of $f(k,\omega)$:

$$f(k,\omega) = \frac{mc}{2\pi^2 e^2 Z}\sigma_\gamma(\omega) \tag{2}$$

Z being the atomic number of the medium. The real part of ε can be expressed as an integral of the imaginary part according to the Kronig's and Kramers' dispersion relation [68]:

$$\begin{aligned} \varepsilon_1 - 1 &= \frac{2}{\pi} P \int_0^\infty \frac{x\varepsilon_2(x)dx}{x^2-\omega^2} \\ &= \frac{2Nc}{\pi Z} P \int_0^\infty \frac{\sigma_\gamma(x)dx}{x^2-\omega^2} \end{aligned} \tag{3}$$

where P indicates the Cauchy principal value. Using these assumptions for the interaction between the projectile and the atom, the following form is obtained for the collision cross-section [69]:

$$\begin{aligned} \frac{d\sigma}{dE} &= -\frac{e^2}{\pi\hbar^2 c^2 n_a} \operatorname{Im}\left[\left(\frac{1}{\beta^2\varepsilon} - 1\right)\ln\frac{2mv^2}{E(1-\beta^2\varepsilon)}\right] \\ &\quad + \frac{2\pi Z e^4}{mv^2 E^2}\int_0^\infty \frac{f(E')}{\varepsilon(E')^2} dE' \end{aligned} \tag{4}$$

where n_a is the number of atoms per cm^3, v velocity of the particle, and $\beta = v/c$. The number of collisions per distance x and the energy E is then

$$\frac{N^2}{dxdE} = n_a \frac{d\sigma}{dE} \tag{5}$$

For the simulation purposes, the number of primary collisions in a unit length with energy loss greater than a certain E is computed and tabulated for several values of the Lorentz factor. This is done by integrating

$$\left(\frac{dN}{dx}\right)_{>E} = \int_E^{E_{max}} n_a \frac{d\sigma}{dE} dE \tag{6}$$

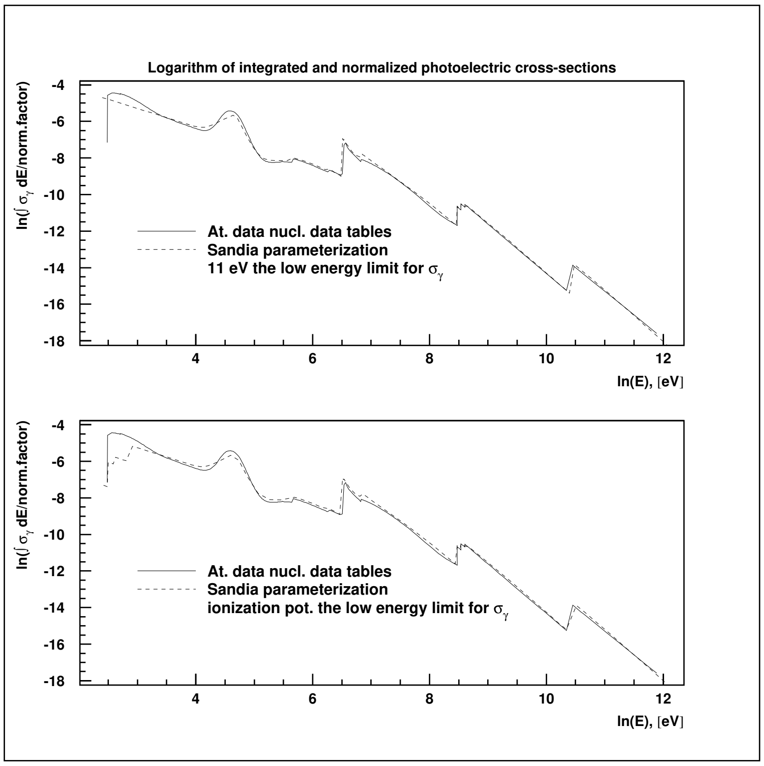


Figure 39: Above, the integrated and normalized cross-section of the gas mixture 70% Xe - 10% CO_2 - 20% CF_4 for two different data sets. The low energy limit of the cross-sections from the Sandia data in chosen to be 11 eV. In the figure below, the low energy limit is the ionization limit for each component of the mixture

2.2 Implementation

A photoabsorption ionization Monte Carlo code provided to us by users has been implemented in GEANT. Tables are prepared at initialization time and sampling from these tables is done each step at running time. An important thing to note is the dependence of the $(dN/dx)_{>E}$ on the photoelectric cross-sections which, when taken from different sources, may differ considerably, especially at low energies.

To fill the tables, the integration of equation 4 has to be done. Before the integration over the energy, the integral over the oscillator strength function $f(k,\omega) \sim \sigma_\gamma(\omega)$ (last term in equation 4) and the dielectric constant (equations 3 and 1) ε have to be computed. These integaration can be done analytically when the parameterized cross-sections are used.

The oscillator strength function should fulfill the Bethe sum rule [51]:

$$\int_0^\infty f(k,\omega) d\omega = 1 \tag{7}$$

To respect this condition, in the PAI model, one has to add some contribution from the excitation processes to low-energy region of the photoelectric cross-sections [68]. In GEANT, the photoelectric cross-section parameterization of the Sandia [19] report is used. In order not to change the tabulated parameterization, the sum rule is taken into account by lowering the energy limit for the integration below the ionization limit. This will not satisy the equation 7 in all cases. In the program, however, the Bethe sum rule is forced by dividing the integrated cross-section from E to ∞ by the value given by the integration over the whole range. In figure 39, the integrated cross-section (oscillator strength function) $\int_E^{E_{max}} \sigma_\gamma / \int_{E_0}^{E_{max}} \sigma_\gamma$ is given for two different sets of photoelectric cross-section data for a gas mixture 70% Xe - 10% CO_2 - 20% CF_4. In the figure below, the continuous line is the result of the integration tabulated from [70] [71] [72] and the dashed line is from the integration of GEANT cross-sections considering the lower limit of the cross-section the ionization energy (12.3 eV for Xe, 11.26 eV for C, 13.6 for O, and 17.42 for F). In the figure above, the

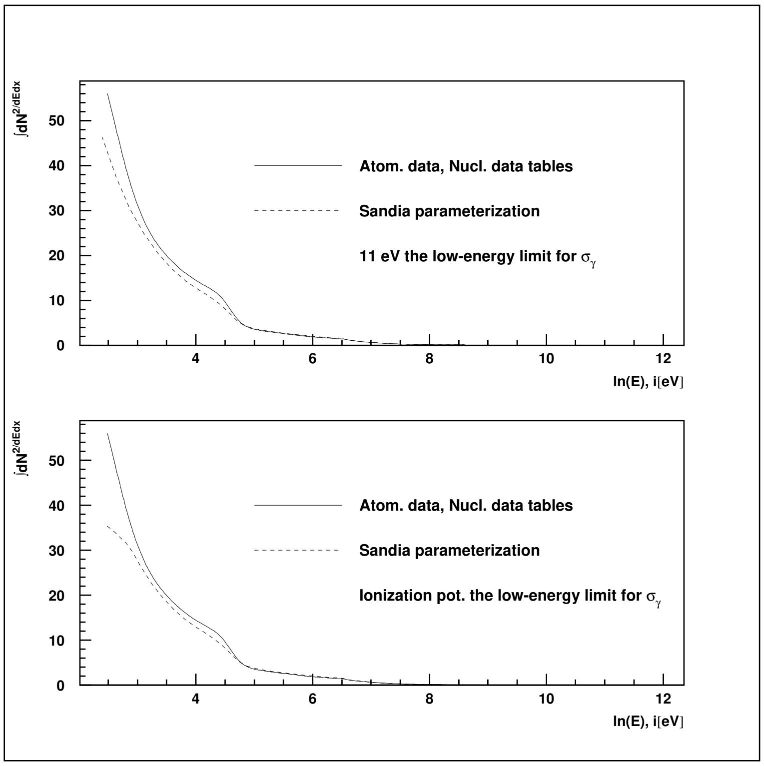


Figure 40: Above, the integrated and normalized cross-section of the gas mixture 70% Xe - 10% CO_2 - 20% CF_4 for two different data sets. The low energy limit of the cross-sections from the Sandia data in chosen to be 11 eV. In the figure below, the low energy limit is the ionization limit for each component of the mixture

continuous line is as before, but for the dashed line, the lower limit of the cross-sections is considered to be 11 eV for all elements of the mixture to compensate for the sum rule.

The imaginary part of the complex dielectric constant can then be computed from the integrated cross-section. To compute the real part of ε, one needs to take the Cauchy principal value which is basically a limiting process cancelling the two infinite contribution around the pole $(x_0 - \delta, x_0 + \delta)$. The derivation of the real part with the Sandia parametrization is shown in detail in the appendix.

Knowing the values of complex dielectric constant and the integrated cross-section, the integration of equation 6 can be done. This is done for several values of Lorentz factor during the initialization time in subroutine `GSTINI`. The value of energy loss is sampled from these tables in subroutine `GSTREN`.

One should remember the dependence of this method on the photoelectric cross section data. Figure 40 shows the results for the two different data sets. The difference in the low-energy part is very significative as the sampling is done from these tables and, according the inverse transverse method, the number of collisions is sampled between the maximum and minimum of the curves in figure 40.

Origin : N.Van Eijndhoven
Revision :
Documentation :

Submitted: 26.07.93
Revised: 16.12.93

Birks' saturation law

1 Subroutines

```
CALL GBIRK (EDEP)
```

`EDEP` (`REAL`) and on output the energy equivalent to the calorimeter response in the current step.

Organic scintillators are usually calibrated with particles whose energy is near the minimum ionisation ($\gamma\beta \approx 4$). However, the energy response of such scintillators for large local energy deposit is attenuated. `GBIRK` returns the energy *detected* by the scintillator in the current step.

2 Method

The phenomenological description of the response attenuation of organic scintillators [73] is known as Birks' law:

$$R = \frac{\Delta E}{1 + C_1\delta + C_2\delta^2} \qquad \delta = \frac{1}{\rho}\frac{dE}{dx} \quad \text{MeV g}^{-1}\text{ cm}^2 \tag{1}$$

The values quoted in [74] for the parameters C_1, C_2 are:

$$C_1 = 0.013 \text{ g MeV}^{-1}\text{ cm}^{-2} \qquad \text{and} \qquad C_2 = 9.6 \times 10^{-6}\text{ g}^2\text{ MeV}^{-2}\text{ cm}^{-4}$$

These values have been measured for various scintillators. If the charge of the particle is greater than one, a better description can be obtained by correcting C_1:

$$C_1' = \frac{7.2}{12.6}C_1 \approx 0.5714 C_1 \tag{2}$$

The values of the parameters of Birks' formula (if defined) are in the `ZEBRA` bank next to the material bank:

`JTM = LQ(JMATE-NTMED)`	pointer to the current tracking medium bank;
`JTMN = LQ(JTM)`	pointer to the *next* bank to the current material one, where the specific tracking medium parameters are stored;
`MODEL = Q(JTMN+27)`	flag controlling the correction (2) for multiply charged particles:
	$\neq$1 correction is not applied;
	=1 correction is applied;
`C1 = Q(JTMN+28)`	first parameter of the Birks' formula;
`C2 = Q(JTMN+29)`	second parameter of the Birks' formula.

These parameters are set via the `GSTPAR` routine, with names `BIRK1`, `BIRK2` and `BIRK3`, respectively. For instance, to define the standard Birks' parameters for tracking medium `ITM` with correction for multiply charged particles, one would have to insert the following piece of code, after the definition of tracking media but before the call to `GPHYSI`

```
      CALL GSTPAR(ITM,'BIRK1',1.)
      CALL GSTPAR(ITM,'BIRK2',0.013)
      CALL GSTPAR(ITM,'BIRK3',9.6E-6)
```

Origin : L.Urbán **Submitted:** 28.05.86
Revision : **Revised:** 16.12.93
Documentation :

Total cross-section and energy loss for bremsstrahlung by e^-/e^+

1 Subroutines

CALL **GBRELA**

GBRELA fills the tables for the energy loss of electrons, positrons and muons due to bremsstrahlung at initialisation time for different materials. The energy binning is set within the array ELOW (common /CGMULO/) in the routine GPHYSI. In the tables, the dE/dx due to bremsstrahlung is summed with that due to the ionisation. For energy loss of electrons and positrons, GBRELA calls the function GBRELE. Following pointers are used:

JMA = LQ(JMATE-I)	pointer to the I^{th} material;
JEL1 = LQ(JMA-1)	pointer to dE/dx for e^-;
JEL1+NEK1	pointer to dE/dx for e^+.

GBRELA is called at initialisation time by GPHYSI.

VALUE = **GBRELE** (ZZ,T,BCUT)

GBRELE calculates the energy loss due to bremsstrahlung of an electron with kinetic energy T in material with atomic number ZZ. It is called by GBRELA and for energies below the cut BCUT it adds the contribution of bremsstrahlung to. Above this cut, the bremsstrahlung process is simulated explicitly (see [PHYS341]) and tabulation of these continuous losses is not needed. GBRELE is called by GBRELA.

VALUE = **GBFLOS** (T,C)

GBFLOS calculates a weight factor for the positron continuous bremsstrahlung energy loss. T is the kinetic energy in GeV of the positron and C is the energy cut for bremsstrahlung (BCUTE). The value is the ratio of the energy loss due to bremsstrahlung of the positron to that of the electron so that: <*positron loss*> = GBFLOS $\times$ <*electron loss*>. GBFLOS is called by GBRELA.

CALL **GBRSGA**

GBRSGA calculates the total cross-section for bremsstrahlung in all materials. It tabulates the mean free path, $\lambda = \frac{1}{\Sigma}$ (in cm) as a function of medium and energy. The energy binning is set within the array ELOW (common /CGMULO/) in the routine GPHYSI. The following pointers are used:

JMA = LQ(JMATE-I)	pointer to the I^{th} material
JBREM = LQ(JMA-9)	pointer to bremsstrahlung cross-sections
JBREM	pointer for e^-
JBREM+NEK1	pointer for e^+
JBREM+2*NEK1	pointer for μ^+/μ^-

GBRSGA is called at initialisation time by GPHYSI.

VALUE = **GBRSGE** (ZZ,T,BCUT)

GBRSGE calculates the total cross-section of bremsstrahlung of an electron with kinetic energy T in material with atomic number ZZ. It is called by GBRSGA. For kinetic energies which are below the cut BCUT or for which bremsstrahlung process is not simulated explicitly (see [PHYS341]) it returns 0.

`VALUE = GBFSIG (T,C)`

GBFSIG calculates a weight factor for the positron discrete (hard) bremsstrahlung cross section. T is the kinetic energy in GeV of the positron and C is the energy cut for bremsstrahlung (BCUTE). The value returned is the ratio of the positron bremsstrahlung cross-section to that of the electron so that: <*positron cross-section*> = GBFSIG $\times$ <*electron cross-section*>. GBFSIG is called by GBRSGA.

2 Method

Let's call $d\sigma(Z, T, k)/dk$ the differential cross-section for production of a photon of energy k by an electron of kinetic energy T in the field of an atom of charge Z, and k_c the energy cut-off below which the soft photons are treated as continuous energy loss (BCUTE in the program). Then the mean value of the energy lost by the electron due to soft photons is

$$E_{Loss}^{brem}(Z, T, k_c) = \int_0^{k_c} k \frac{d\sigma(Z, T, k)}{dk} dk \tag{1}$$

whereas the total cross-section for the emission of a photon of energy larger than k_c is

$$\sigma_{brem}(Z, T, k_c) = \int_{k_c}^{T} \frac{d\sigma(Z, T, k)}{dk} dk \tag{2}$$

Many theories of the bremsstrahlung process exist, each with its own limitations and regions of applicability. Perhaps the best synthesis of these theories can be found in the paper of S.M. Seltzer and M.J. Berger [75]. 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 [12, 35] together with the Midgal corrections [54, 76]. 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 – 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 $k/T \leq 10^{-4}$.)

2.1 Parameterisation of energy loss and total cross-section

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 chosen the parameterisations:

$$\sigma(Z, T, k_c) = \frac{Z(Z + \xi_\sigma)(T + m)^2}{T(T + 2m)} [\ln(T/k_c)]^\alpha F_\sigma(Z, X, Y) \quad \text{(barn)} \tag{3}$$

and

$$E_{Loss}^{brem}(Z, T, k_c) = \frac{Z(Z + \xi_l)(T + m)^2}{(T + 2m)} \left[\frac{k_c C_M}{T}\right]^\beta F_l(Z, X, Y) \quad \text{(GeV barn)} \tag{4}$$

where m is the mass of the electron,

$$\begin{array}{ll} X = \ln(E/m), \quad Y = \ln(v_\sigma E/k_c) & \text{for the total cross-section } \sigma \\ X = \ln(T/m), \quad Y = \ln(k_c/v_l E) & \text{for the energy loss } E_{Loss}^{brem} \end{array}$$

with $E = T + m$. The constants $\xi_\sigma,\ \xi_l,\ \alpha,\ \beta,\ v_\sigma,\ v_l$ are parameters to be fitted.

$$C_M = \frac{1}{1 + \dfrac{n r_0 \lambda\!\!\!^{-2} (T + m)^2}{\pi k_c^2}}$$

is the Midgal correction factor, with

r_o classical electron radius;
$\lambda\!\!\!^{-}$ reduced electron Compton wavelength;
n electron density in the medium.

The factors $(T+m)^2/T(T+2m)$ and $(T+m)^2/(T+2m)$ come from the scaled cross-section computed by Seltzer and Berger:

$$f(k/T) = \frac{\beta^2}{Z^2} k \frac{d\sigma}{dk} = \frac{T(T+2m)}{(T+m)^2 Z^2} k \frac{d\sigma}{dk}$$

The functions $F_i(Z, X, Y)$ $(i = \sigma, l)$ have the form

$$F_i(Z, X, Y) = F_{i0}(X, Y) + Z F_{i1}(X, Y) \tag{5}$$

where $F_{ij}(X, Y)$ are polynomials of the variables X, Y

$$\begin{aligned} F_{i0}(X,Y) &= (C_1 + C_2 X + \ldots + C_6 X^5) + (C_7 + C_8 X + \ldots + C_{12} X^5) Y \\ &\quad + (C_{13} + C_{14} X + \ldots + C_{18} X^5) Y^2 + \ldots + (C_{31} + C_{32} X + \ldots + C_{36} X^5) Y^5 \\ &\qquad\qquad Y \leq 0 \\ &= (C_1 + C_2 X + \ldots + C_6 X^5) + (C_7 + C_8 X + \ldots + C_{12} X^5) \\ &\quad + (C_{37} + C_{38} X + \ldots + C_{42} X^5) Y^2 + \ldots + (C_{55} + C_{56} X + \ldots + C_{60} X^5) Y^5 \\ &\qquad\qquad Y > 0 \end{aligned} \tag{6}$$

$$\begin{aligned} F_{i1}(X,Y) &= (C_{61} + C_{62} X + \ldots + C_{65} X^4) + (C_{66} + C_{67} X + \ldots + C_{70} X^4) Y \\ &\quad + (C_{71} + C_{72} X + \ldots + C_{75} X^4) Y^2 + \ldots + (C_{81} + C_{82} X + \ldots + C_{85} X^4) Y^4 \\ &\qquad\qquad Y \leq 0 \end{aligned} \tag{7}$$

$$\begin{aligned} &= (C_{61} + C_{62} X + \ldots + C_{65} X^4) + (C_{66} + C_{67} X + \ldots + C_{70} X^4) Y \\ &\quad + (C_{86} + C_{87} X + \ldots + C_{90} X^4) Y^2 + \ldots + (C_{96} + C_{97} X + \ldots + C_{100} X^4) Y^4 \\ &\qquad\qquad Y > 0 \end{aligned} \tag{8}$$

$F_{ij}(X, Y)$ denotes in fact a function constructed from two polynomials

$$F_{ij}(X,Y) = \begin{cases} F_{ij}^{neg}(X,Y) & \text{for} \quad Y \leq 0 \\ F_{ij}^{pos}(X,Y) & \text{for} \quad Y > 0 \end{cases}$$

where the polynomials F_{ij} fulfil the conditions

$$F_{ij}^{neg}(X,Y)_{Y=0} = F_{ij}^{pos}(X,Y)_{Y=0} \qquad \left(\frac{\partial F_{ij}^{neg}}{\partial Y}\right)_{Y=0} = \left(\frac{\partial P_{ij}^{pos}}{\partial Y}\right)_{Y=0}$$

We have computed 4000 "data points" in the range

$$Z = 6; 13; 29; 47; 74; 92 \qquad 10 \text{ keV} \leq T \leq 10 \text{ TeV} \qquad 10 \text{ keV} \leq k_c \leq T$$

and we have performed a least-squares fit to determine the parameters.

The values of the parameters (ξ_σ, α, v_σ, C_i for σ and ξ_l, β, V_l, C for E_{Loss}^{brem}) can be found in the `DATA` statement within the functions `GBRSGE` and `GBRELE` which compute the formula (3) and (4) respectively.

The errors of the parameterisations (3) and (4) can be estimated as

$$\frac{\Delta\sigma}{\sigma} = \begin{cases} 12-15\% & \text{for} \quad T \leq 1MeV \\ \leq 5-6\% & \text{for} \quad 1MeV < T \leq 10TeV \end{cases}$$

$$\frac{\Delta E^{brem}_{Loss}}{E^{brem}_{Loss}} = \begin{cases} 10-15\% & \text{for} \quad T \leq 1MeV \\ 5-6\% & \text{for} \quad 1MeV < T \leq 100GeV \\ 10\% & \text{for} \quad 100GeV < T \leq 10TeV \end{cases}$$

We have performed a fit to the "data" without the Midgal corrections, too. In this case we used the data of Seltzer and Berger without any correction for $T \leq 10$ GeV and we used the Bethe-Heitler cross-section for $T \geq 10$ GeV. The parameterised forms of the cross-section and energy loss are the same as they were 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 parameterisation can be activated using the `PATCHY` switch `+USE,BETHE`. (The two parameterisations give different results for high electron energy.)

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 mean free path for discrete bremsstrahlung is tabuled at initialisation time as a function of the medium and of the energy by routine `GBRSGA` (see `JMATE` data structure).

2.2 Corrections for $\mathrm{e}^-/\mathrm{e}^+$ differences

The radiative energy loss for electrons or positrons is:

$$\begin{aligned} -\frac{1}{\rho}\left(\frac{dE}{dx}\right)^{\pm}_{rad} &= \frac{N_{Av}\alpha r_e^2}{A}(T+m)Z^2\Phi^{\pm}_{rad}(Z,T) \\ \Phi^{\pm}_{rad}(Z,T) &= \frac{1}{\alpha r_e^2 Z^2(T+m)}\int_0^T k\frac{d\sigma^{\pm}}{dk}dk \end{aligned}$$

Reference [77] says that:

"The differences between the radiative loss of positrons and electrons are considerable and cannot be disregarded.

[...] The ratio of the radiative energy loss for positrons to that for electrons obeys a simple scaling law, [...] is a function only of the quantity T/Z^2"

In other words:

$$\eta = \frac{\Phi^{+}_{rad}(Z,T)}{\Phi^{-}_{rad}(Z,T)} = \eta\left(\frac{T}{Z^2}\right)$$

The authors have calculated this function in the range $10^{-7} \leq \frac{T}{Z^2} \leq 0.5$ (here the kinetic energy T is expressed in MeV). Their *data* can be fairly accurately reproduced using a parametrisation:

$$\eta = \begin{cases} 0 & \text{if} \quad x \leq -8 \\ \frac{1}{2}+\frac{1}{\pi}\arctan\left(a_1x+a_3x^3+a_5x^5\right) & \text{if} \quad -8 < x < 9 \\ 1 & \text{if} \quad x \geq 9 \end{cases}$$

where:

$$
\begin{aligned}
x &= \log\left(C\frac{T}{Z^2}\right) \text{ (T in GeV)} \\
C &= 7.5221 \times 10^6 \\
a_1 &= 0.415 \\
a_3 &= 0.0021 \\
a_5 &= 0.00054
\end{aligned}
$$

This e^-/e^+ energy loss difference is not a pure low-energy phenomenon (at least for high Z), as it can be seen from Tables 1.2, 1.3 and 1.4.

$\frac{T}{Z^2}(GeV)$	T	η	$\left(\frac{rad.\ loss}{total\ loss}\right)_{e^-}$
10^{-9}	$\sim 7keV$	~ 0.1	$\sim 0\%$
10^{-8}	$67keV$	~ 0.2	$\sim 1\%$
2×10^{-7}	$1.35MeV$	~ 0.5	$\sim 15\%$
2×10^{-6}	$13.5MeV$	~ 0.8	$\sim 60\%$
2×10^{-5}	$135.MeV$	~ 0.95	$> 90\%$

Table 1.2: ratio of the e^-/e^+ radiative energy loss in lead (Z=82).

The scaling holds for the ratio of the total radiative energy losses, but it is significantly broken for the photon spectrum in the screened case. In case of a point Coulomb charge the scaling would hold also for the spectrum. The scaling can be expressed by:

$$\frac{\Phi^+}{\Phi^-} = \eta\left(\frac{T}{Z^2}\right) \qquad\qquad \frac{\frac{d\sigma^+}{dk}}{\frac{d\sigma^-}{dk}} = \text{does not scale}$$

If we consider the photon spectrum from bremsstrahlung reported in [77] we see that:

$$\frac{d\sigma^\pm}{dk} = S^\pm\left(\frac{k}{T}\right) \qquad \frac{S^+(k)}{S^-(k)} \le 1 \qquad S^+(1) = 0 \qquad S^-(1) > 0$$

We further assume that:

$$\frac{d\sigma^+}{dk} = f(\epsilon)\frac{d\sigma^-}{dk} \qquad\qquad \epsilon = \frac{k}{T} \tag{9}$$

In order to satisfy approximately the scaling law for the ratio of the total radiative energy loss, we require for $f(\epsilon)$:

$$\int_0^1 f(\epsilon)d\epsilon = \eta \tag{10}$$

From the photon spectra we require:

$$\left.\begin{array}{l} f(0) = 1 \\ f(1) = 0 \end{array}\right\} \qquad \text{for all } Z, T \tag{11}$$

T(MeV)	C			Pb		
	ΔE_l^0	ΔE_l	$\Delta\sigma_l$	ΔE_l^0	ΔE_l	$\Delta\sigma_l$
0.02	-2.86	-2.86	+52.00	-4.89	-4.69	+99.80
0.1	-0.33	-0.33	+21.10	-0.52	-0.47	+81.08
1	+0.07	+0.07	+6.49	+0.11	+0.11	+48.99
10	0.00	0.00	+1.75	0.00	+0.01	+23.89
10^2	0.00	0.00	0.00	0.00	0.00	+9.00
10^3	0.00	0.00	0.00	0.00	0.00	+2.51
10^4	0.00	0.00	0.00	0.00	0.00	+0.00

$$\Delta E_l = 100\frac{E_l^- - E_l^+}{E_l^-}\% \qquad \text{and} \qquad \Delta\sigma_l = 100\frac{\sigma_l^- - \sigma_l^+}{\sigma_l^-}\%$$

Table 1.3: Difference in the energy loss and bremsstrahlung cross-section for e^-/e^+ in Carbon and Lead with a cut for γ and $e^\pm$ of 10keV. ΔE_l^0 is the value without the correction for the difference e^-/e^+.

T(MeV)	C			Pb		
	ΔE_l^0	ΔE_l	$\Delta\sigma_l$	ΔE_l^0	ΔE_l	$\Delta\sigma_l$
2	+4.19	+4.21	+7.29	+4.47	+6.88	+61.78
10	+0.87	+0.87	+1.93	+0.87	+1.14	+26.29
10^2	+0.08	+0.08	0.00	+0.06	+0.06	+9.10
10^3	0.00	0.00	0.00	0.00	0.00	+2.42
10^4	0.00	0.00	0.00	0.00	0.00	+0.00

$$\Delta E_l = 100\frac{E_l^- - E_l^+}{E_l^-}\% \qquad \text{and} \qquad \Delta\sigma_l = 100\frac{\sigma_l^- - \sigma_l^+}{\sigma_l^-}\%$$

Table 1.4: Difference in the energy loss and bremsstrahlung cross-section for e^-/e^+ in Carbon and Lead with a cut for γ and $e^\pm$ of 1MeV. ΔE_l^0 is the value without the correction for the difference e^-/e^+.

We have chosen a simple function f:

$$f(\epsilon) \quad = \quad C(1-\epsilon)^\alpha \qquad\qquad C, \alpha > 0 \tag{12}$$

from the conditions (10), (11) we get:

$$\begin{aligned} C &= 1 \\ \alpha &= \frac{1}{\eta} - 1 \qquad\qquad (\alpha > 0 \text{ because } \eta < 1) \\ f(\epsilon) &= (1-\epsilon)^{\frac{1}{\eta}-1} \end{aligned}$$

We have defined weight factors F_l and F_σ for the positron continuous energy loss and discrete bremsstrahlung cross section:

$$F_l = \frac{1}{\epsilon_0}\int_0^{\epsilon_0} f(\epsilon)d\epsilon \qquad\qquad F_\sigma = \frac{1}{1-\epsilon_0}\int_{\epsilon_0}^1 f(\epsilon)d\epsilon \tag{13}$$

$$100\frac{E^{+}_{dep} - E^{-}_{dep}}{\sqrt{\sigma^2_+ + \sigma^2_-}}(\%)$$

Depth	C		Pb	
(X_0 units)	No $e^{\pm}$ diff	$e^{\pm}$ diff	No $e^{\pm}$ diff	$e^{\pm}$ diff
0.5	-11.7	-13.0	-0.8	-3.9
1.0	-5.3	-4.9	-1.0	-4.1
1.5	+7.3	+8.0	-1.4	-3.5
2.0	+7.1	+5.3	-0.7	-0.0
2.5	+4.9	+4.3	+1.7	+3.6
3.0	+4.8	+4.1	+1.1	+4.3
3.5	+3.3	+2.7	+2.7	+3.1
4.0	+3.6	+5.3	+2.9	+3.0
4.5	+1.7	+2.8	+0.5	+2.3
5.0	+3.4	+3.5	-1.9	+1.8

Table 1.5: Difference in the shower development for e^-/e^+ in Carbon and Lead. *No diff* refers to the value without the correction for the difference e^-/e^+.

where $\epsilon_0 = \frac{k_c}{T}$ and k_c is the photon cut BCUTE. In this scheme the positron energy loss and discrete bremsstrahlung can be calculated as:

$$\left(-\frac{dE}{dx}\right)^+ = F_l\left(-\frac{dE}{dx}\right)^- \qquad\qquad \sigma^+_{brems} = F_\sigma \sigma^-_{brems}$$

As in this approximation the photon spectra are identical, the same SUBROUTINE is used for generating e^-/e^+ bremsstrahlung. The following relations hold:

$$\begin{aligned} F_\sigma &= \eta(1-\epsilon_0)^{\frac{1}{\eta}-1} < \eta \\ \epsilon_0 F_l + (1-\epsilon_0)F_\sigma &= \eta && \text{from the def (13)} \\ \Rightarrow F_l &= \eta\frac{1-(1-\epsilon_0)^{\frac{1}{\eta}})}{\epsilon_0} > \eta\frac{1-(1-\epsilon_0)}{\epsilon_0} = \eta && \Rightarrow \begin{cases} F_l > \eta \\ F_\sigma < \eta \end{cases} \end{aligned}$$

which is consistent with the spectra.

The effect of this e^-/e^+ bremsstrahlung difference can be also seen in e.m. shower development, when the primary energy is not too high. An example can be found in table 1.5.

Origin : G.N.Patrick, L.Urbán **Submitted:** 26.10.84
Revision : **Revised:** 16.12.93
Documentation :

Simulation of discrete bremsstrahlung by electrons

1 Subroutines

CALL **GBREME**

GBREME generates a bremsstrahlung photon from an electron as a discrete process. The photon energy is sampled from a parameterisation of the bremsstrahlung cross-section of Seltzer and Berger [75] for electron energies below 10 GeV, and from the screened Bethe-Heitler cross-section above 10 GeV. Midgal corrections are applied in both cases. The angular distribution of the photon is calculated by the function GBTETH.

Input : via common block /GCTRAK/

Output: via common block /GCKING/.

GBREME is called from the tracking routine GTELEC when the parent electron reaches a radiation point during tracking.

THETA = **GBTETH** (ENER,PARTM,EFRAC)

THETA	(REAL) angle of the radiated photon or $e^{\pm}$ pair;
ENER	(REAL) energy of the particle;
PARTM	(REAL) mass of the particle;
EFRAC	(REAL) ratio between the energy of the photon and the energy of the particle.

GBTETH calculates the angular distribution of the e^+e^--pair in photon pair production and of the emitted photon in μ and $e^{\pm}$ bremsstrahlung. GBTETH is called by GBREME.

2 Method

The photon energy is sampled according to the Seltzer and Berger bremsstrahlung spectrum [75]. Seltzer and Berger have calculated the spectra for materials with atomic numbers Z = 6,13,29,47,74,92 in the electron (kinetic) energy range 1 keV - 10 GeV. Their tabulated results have been used as input in a parametrising-fitting procedure. The functional form of the parameterisation for the quantity:

$$S(x) = Ck\frac{d\sigma}{dk}$$

can be written as

$$S(x) = \begin{cases} (1 - a_h\epsilon)F_1(\delta) + b_h\epsilon^2 F_2(\delta) & T \geq 1MeV \\ 1 + a_l x + b_l x^2 & T < 1MeV \end{cases} \tag{1}$$

where:

C	normalisation constant	k	photon energy
T, E	kinetic and total energy of the primary electron	$x = \dfrac{k}{T}$	
$\epsilon = \dfrac{k}{E} = x\dfrac{T}{E}$			

The $F_i(\delta)$ screening functions depend on the screening variable:

$$
\begin{aligned}
\delta &= \frac{136 m_e}{Z^{1/3}E}\frac{\epsilon}{1-\epsilon} \\
F_1(\delta) &= F_0(42.392 - 7.796\delta + 1.961\delta^2 - F) && \delta \leq 1 \\
F_2(\delta) &= F_0(41.734 - 6.484\delta + 1.250\delta^2 - F) && \delta \leq 1 \\
F_1(\delta) &= F_2(\delta) = F_0(42.24 - 8.368\ln(\delta + 0.952) - F) && \delta > 1 \\
F_0 &= \frac{1}{42.392 - F} \\
F &= 4\ln Z - 0.55(\ln Z)^2
\end{aligned}
$$

$a_{h,l}$ and $b_{h,l}$ are parameters to be fitted.

The 'high energy' (T > 1 MeV) formula comes from the Coulomb-corrected, sceened Bethe-Heitler formula (see e.g. [78, 10, 12]). However, there are two things in eq. (1) which make a difference:

1. a_h, b_h depend on T and on the atomic number Z (in the case of the Bethe-Heitler spectrum $a_h = 1$, $b_h = 0.75$);
2. the function F is not the same than that in the Bethe-Heitler cross-section, this function gives a better behaviour in the high frequency limit, i.e. when $k \rightarrow T$ ($x \rightarrow 1$).

The T and Z dependence of the parameters are described by the equations:

$$
\begin{aligned}
a_h &= 1 + \frac{a_{h1}}{u} + \frac{a_{h2}}{u^2} + \frac{a_{h3}}{u^3} \\
b_h &= 0.75 + \frac{b_{h1}}{u} + \frac{b_{h2}}{u^2} + \frac{b_{h3}}{u^3} \\
a_l &= a_{l0} + a_{l1}u + a_{l2}u^2 \\
b_l &= b_{l0} + b_{l1}u + b_{l2}u^2
\end{aligned}
$$

with

$$u = \ln\left(\frac{T}{m_e}\right)$$

the $a_{hi}, b_{hi}, a_{li}, b_{li}$ parameters are polynomials of second order in the variable:

$$v = [Z(Z+1)]^{1/3}$$

It can be seen relatively easily that for the limiting case $T \rightarrow \infty$, $a_h \rightarrow 1, b_h \rightarrow 0.75$, so eq. (1) gives the Bethe-Heitler cross section.

There are altogether 36 linear parameter in the formulae , their values are given in GBREME. The parameterisation reproduces the Seltzer-Berger tables within a few % (2-3 % on average, the maximum error being less than 10-12 %), the tables, on the other hand, agree well with the experimental data and theoretical (low- and high-energy) results (less than 10 % below 50 MeV, less than 5 % above 50 MeV).

Apart from the normalisation the cross section differential in photon energy can be written as:

$$\frac{d\sigma}{dk} = \frac{1}{\ln\frac{1}{x_c}}\frac{1}{x}g(x) = \frac{1}{\ln\frac{1}{x_c}}\frac{1}{x}\frac{S(x)}{S_{max}}$$

where $x_c = k_c/T$, k_c is the photon cut-off energy below which the bremsstrahlung is treated as a continuous energy loss (this cut is BCUTE in the program). Using this decomposition of the cross section and two random numbers r_1, r_2 uniformly distributed in $]0,1[$, the sampling of x is done as follows:

1. sample x from

$$\frac{1}{\ln \frac{1}{x_c}} \frac{1}{x} \qquad \text{setting} \qquad x = e^{r_1 \ln x_c}$$

2. calculate the rejection function $g(x)$ and:
   - if $r_2 > g(x)$ reject x and go back to 1;
   - if $r_2 \leq g(x)$ accept x.

To apply the Migdal correction [76] all it has to be done is to multiply the rejection function by the Migdal correction factor:

$$C_M(\epsilon) = \frac{1 + C_0/\epsilon_c^2}{1 + C_0/\epsilon^2}$$

where

$$C_0 = \frac{n r_0 \lambda\!\!\!^{-2}}{\pi}, \qquad \epsilon_c = \frac{k_c}{E}$$

n electron density in the medium
r_0 classical electron radius
$\lambda\!\!\!^{-}$ reduced Compton wavelength of the electron.

This correction decreases the cross-section for low photon energy.

After the successful sampling of ϵ, GBREME generates the polar angles of the radiated photon with respect to the parent electron's momentum. It is difficult to find in the literature simple formulas for this angle. For example the double differential cross section reported by Tsai [13, 14] is the following:

$$\begin{aligned}
\frac{d\sigma}{dk d\Omega} &= \frac{2\alpha^2 e^2}{\pi k m^4} \left\{ \left[\frac{2\epsilon - 2}{(1+u^2)^2} + \frac{12u^2(1-\epsilon)}{(1+u^2)^4} \right] Z(Z+1) \right. \\
&\quad \left. + \left[\frac{2 - 2\epsilon - \epsilon^2}{(1+u^2)^2} - \frac{4u^2(1-\epsilon)}{(1+u^2)^4} \right] \left[X - 2Z^2 f_c((\alpha Z)^2) \right] \right\} \\
u &= \frac{E\theta}{m} \\
X &= \int_{t_{min}}^{m^2(1+u^2)^2} \left[G_Z^{el}(t) + G_Z^{in}(t) \right] \frac{t - t_{min}}{t^2} dt \\
G_Z^{el,in}(t) & \quad \text{atomic form factors} \\
t_{min} &= \left[\frac{k m^2 (1+u^2)}{2E(E-k)} \right]^2 = \left[\frac{\epsilon m^2 (1+u^2)}{2E(1-\epsilon)} \right]^2
\end{aligned}$$

This distribution is complicated to sample, and it is anyway only an approximation to within few percent, if nothing else, due to the presence of the atomic form-factors. The angular dependence is contained in the variable $u = E\theta m^{-1}$. For a given value of u the dependence of the shape of the function on Z, E, $\epsilon = k/E$ is very weak. Thus, the distribution can be approximated by a function

$$f(u) = C \left(u e^{-au} + d u e^{-3au} \right) \tag{2}$$

where

$$C = \frac{9a^2}{9 + d} \qquad a = 0.625 \qquad d = 0.13 \left(0.8 + \frac{1.3}{Z} \right) \left(100 + \frac{1}{E} \right) (1 + \epsilon)$$

where E is in GeV. While this approximation is good at high energies, it becomes less accurate around few MeV. However in that region the ionisation losses dominate over the radiative losses.

The sampling of the function $f(u)$ can be done in the following way (r_i, $i = 1, 2, 3$ are uniformly distributed random numbers in [0,1]):

$$P = \int_{u_{max}}^{\infty} f(u)\, du$$

E (MeV)	P(%)
0.511	3.4
0.6	2.2
0.8	1.2
1.0	0.7
2.0	< 0.1

Table 1.6: Angular sampling efficiency

1. Choose between ue^{-au} and due^{-3au}:

$$b = \begin{cases} a & \text{if} \quad r_1 < 9/(9+d) \\ 3a & \text{if} \quad r_1 \geq 9/(9+d) \end{cases}$$

2. Sample ue^{-bu}:

$$u = -\frac{\log(r_2 r_3)}{b}$$

3. check that:

$$u \leq u_{max} = \frac{E\pi}{m}$$

   otherwise go back to 1.

The probability of failing in the last test is reported in table 1.6.

The function $f(u)$ can be used also to describe the angular distribution of the photon in μ bremsstrahlung and to describe the angular distribution in photon pair production.

The azimuthal angle, Φ, is generated isotropically. This information is used to calculate the momentum vector of the radiated photon, to transform it to the `GEANT` coordinate system and to store the result into common block `/GCKING/`. Also, the momentum of the parent electron is updated.

2.1 Restrictions

1. Target materials composed of compounds or mixtures are treated identically to elements using the effective atomic number Z calculated in `GSMIXT` (this is not correct when computing the mean free path!).

Origin : L.Urbán
Revision : G.Azuelos
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Total cross-section for e^+e^- annihilation

1 Subroutines

CALL **GANNII**

GANNII tabulates the mean free path at initialisation as a function of the medium and of the energy (see JMATE data structure). The energy binning is set within the array ELOW (common block /GCMULO/) in the routine GPHYSI. GANNII is called from GPHYSI.

2 Method

For the annihilation into two photons the cross-section formula of Heitler is used. The total cross-section is given by [12, 79]

$$\begin{aligned} \sigma_{2\gamma}(Z,E) &= \frac{Z\pi r_0^2}{\gamma+1}\left[\frac{\gamma^2+4\gamma+1}{\gamma^2-1}\ln\left(\gamma+\sqrt{\gamma^2-1}\right)-\frac{\gamma+3}{\sqrt{\gamma^2-1}}\right] \quad (1)\\ r_0 &= \text{classical electron radius} \end{aligned}$$

For compounds, the cross-section is calculated using an effective atomic number as explained in [PHYS010].

Positrons can annihilate in a single photon if the electron with which it interacts is bound to a nucleus. The total cross-section for such a process is

$$\begin{aligned} \sigma_{1\gamma}(Z,E) &= 4\pi r_0^2\alpha^4\frac{Z^5}{\gamma\beta(\gamma+1)^2}\left[\gamma^2+\frac{2}{3}\gamma+\frac{4}{3}+\frac{\gamma+2}{\gamma\beta}\ln(\gamma+\gamma\beta)\right] \quad (2)\\ \alpha &= \text{fine structure constant} \end{aligned}$$

In the derivation of this formula, only the interactions with the K-shell electrons are taken into account. As the cross-section depends on Z^5, a special value of Z_{eff} is computed in GPROBI:

$$Z_{eff} = \sum_i \frac{p_i}{A_i} Z^5 \quad (3)$$

The notation of [PHYS010] is used.

The total cross-section for the positron annihilation is $\sigma = \sigma_{2\gamma} + \sigma_{1\gamma}$. The value of $\sigma_{1\gamma}$ is at it largest for heavy materials, for example, it is ~ 20 % of σ for a positron of 440 keV of kinetic energy in lead. For lower and higher energies the probability is lower.

Origin : L.Urbán
Revision : G.Azuelos
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Simulation of e^+e^-annihilation

1 Subroutines

CALL **GANNI**

GANNI generates the annihilation of a positron into either one or two photons. It uses the following input and output:

input: via common /GCTRAK/
output: via common /GCKING/

The routine is called automatically from the tracking routine GTELEC, when the positron reaches its interaction point during the tracking.

CALL **GANNIR**

GANNIR generates the positron annihilation at rest. It is called from the tracking routine GTELEC, if the positron energy is below the cut-off energy CUTELE in common block /GCCUTS/.

2 Method

The type of annihilation is sampled from the total cross-sections for the annihilation into two photons and into one photon (see section [PHYS350]).

Annihilation into two photons

The differential cross-section of the two-photon positron-electron annihilation can be written as [79, 12]:

$$\frac{d\sigma(Z,\epsilon)}{d\epsilon} = m\,a[S(a\epsilon) + S(a(1-\epsilon))] \tag{1}$$

where m is the electron mass Z is the atomic number of the material. If we indicate with E the initial energy of the positron, with r_0 the classical electron radius and with k the energy of the less energetic photon generated, we have:

$$\gamma = \frac{E}{m} \qquad a = \gamma + 1$$
$$\epsilon = \frac{k}{E+m} \qquad S(x) = C_1\left[-1 + \frac{C_2}{x} - \frac{1}{x^2}\right]$$
$$C_1 = \frac{Z\pi r_0^2}{a(E-m)} \qquad C_2 = a + \frac{e\gamma}{a}$$

The kinematical limits for the variable ϵ are:

$$\epsilon_0 = \frac{1}{a+\sqrt{\gamma^2-1}} \le \epsilon \le \frac{1}{2}$$

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 eliminated from the formula. Having done this, the differential cross-section can be decomposed (apart from the normalisation) as:

$$\frac{d\sigma}{d\epsilon} = \underbrace{\frac{1}{\ln\frac{1-\epsilon_0}{\epsilon_0}}\frac{1}{\epsilon}}_{f(\epsilon)}\underbrace{\frac{(a^2+2a-2) - a^2\epsilon - \frac{1}{\epsilon}}{a^2-2}}_{g(\epsilon)} \tag{2}$$

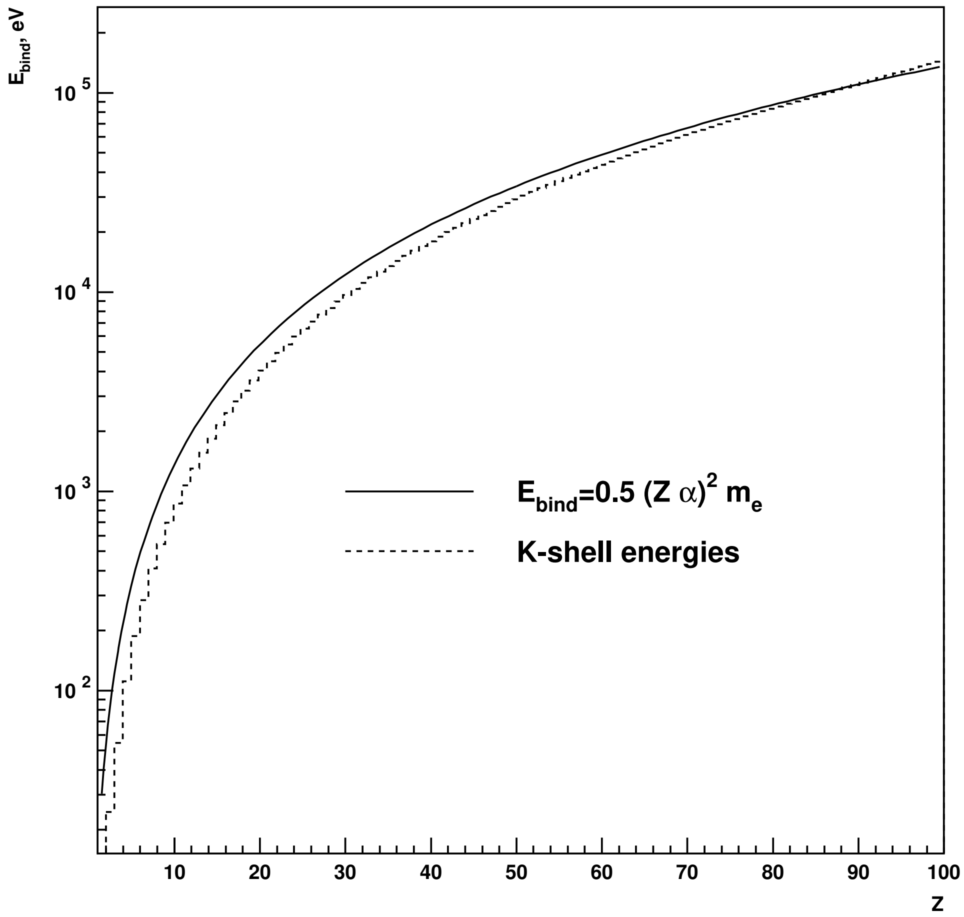


Figure 41: Comparison between the K-shell binding energies given by the expression in the text and the tabulated values.

Using the expression (2) with random numbers $r_i \in]0,1[, i = 1,2$, the secondary photon energy is sampled by the following steps:

1. sample ϵ from $f(\epsilon)$:

$$\epsilon = \epsilon_0 \exp\left[\ln\left(\frac{1-\epsilon_0}{\epsilon_0}\right) r_1\right] \tag{3}$$

2. compute the rejection function $g(\epsilon)$ and
   (a) if $r_2 \leq g(\epsilon)$ accept ϵ
   (b) if $r_1 > g(\epsilon)$ go back to 1.

After the successful sampling of ϵ, the photon energy is computed as

$$k = (E + m)\epsilon \tag{4}$$

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. With this information, the momentum vector of both photons can be calculated and transformed into the `GEANT` coordinate system.

The routine GANNIR treates the special case when a positron falls below the cut-off energy (CUTELE in common block /GCCUTS/) before annihilating. In this case, it is assumed that the positron comes to rest before annihilating. GANNIR generates two photons with energy $k = m$. The angular distribution is isotropic.

Annihilation into one photon

The generated photon is assumed to be collinear with the positron. Its energy will be

$$k = E + m_e - E_{bind} \tag{5}$$

where E_{bind} is the binding energy of the K-shell electron. It can be estimated as follows

$$E_{bind} = 0.5(Z\,\alpha)^2\,m_e \tag{6}$$

where α is the fine stucture constant. The comparison of this expression with the experimental data from [80] is shown in figure 41.

3 Restrictions

1. The annihilation processes producing three or more photons are ignored, because these processes are negligible compared to the annihilation into two photons [12, 54];
2. In calculating the process it is assumed that the atomic electron initially is free and at rest. This is the usual assumption used in shower programs [12];

Origin : I.Gavrilenko
Revision :
Documentation : K.Lassila-Perini, I.Gavrilenko

Submitted: 17.11.93
Revised: 16.12.93

Synchrotron radiation

1 Subroutines

```
CALL GSYNC
```

`GSYNC` is called for electron and positrons when the flag `ISYNC` is set > 0. It computes the magnetic field and calls `GSYNGE` for the generation of the synchrotron radiation photons. It reads them in the `GEANT` stack. `GSYNC` is called from `GTELEC`.

```
CALL GSYNGE (GAMMA,AMASS,FLDT,STEP,NTR,ETR,STR)
```

`GSYNGE` generates the synchrotron radiation for an e$^-$ or e$^+$ with the Lorentz factor `GAMMA` in the magnetic field `FLDT` and when the step length is `STEP`. The number of the secondary photons is `NTR`, their energy is in the array `ETR` and their position along the step is in `STR`. `GSYNGE` is called by `GSYNC`.

2 Method

A charged particle in a magnetic field emits radiation. The number of photons emitted in a second is

$$\frac{d^2N}{d\varepsilon\, dt} = \frac{15\sqrt{3}}{8}\frac{P}{\varepsilon_c} \tag{1}$$

where ε_c is the critical photon energy (the median of the energy spectrum) and P the total radiated power:

$$\varepsilon_c = \frac{3}{2}\hbar c\frac{\gamma^3}{\rho} \tag{2}$$

$$P = \frac{2}{3}\frac{e^2 c}{\rho^2}\beta^4\gamma^4 \tag{3}$$

γ is the e$^-$/e$^+$ Lorentz factor E_{tot}/m_e and $\beta = 1 - 1/\gamma^2$. ρ is the curvature caused by the magnetic field. For more detailed derivation of these equations, see [81]. The velocity of the particle being βc, the number of photons per meter is

$$\begin{aligned}\frac{d^2N}{d\varepsilon\, dx} &= \frac{d^2N}{d\varepsilon\, dt}\frac{1}{\beta c}\\ &= \frac{5\sqrt{3}}{6}\frac{e^2}{\hbar c}\beta^3\gamma\frac{1}{\rho}\\ &= \frac{5}{2\sqrt{3}}\,\alpha\,\frac{\gamma}{\rho} \approx 0.01053\,\frac{\gamma}{\rho}\end{aligned} \tag{4}$$

The fine structure constant $\alpha = e^2/(\hbar c)$ and it is assumed that $\beta^3 \approx 1$.

The curvature in a magnetic field B which has a component $B_\perp$ transversal to the particle velocity can be computed [55]

$$\rho = \frac{p}{0.3B_\perp} \tag{5}$$

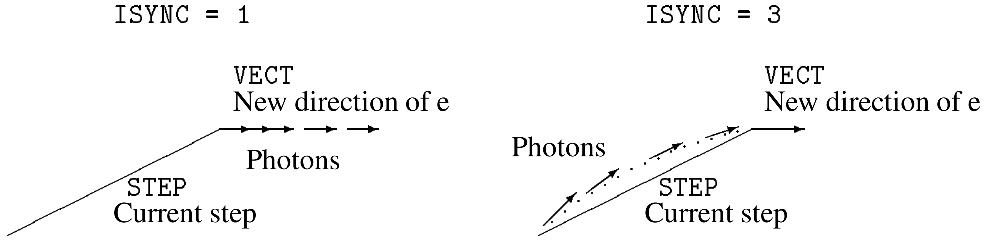


Figure 42: The point where the synchrotron radiation photon is generated. The figure on the left describes the situation when `ISYNC = 1`, and the one on the right when `ISYNC = 3`. The little arrows are the photons and `STEP` is the step taken by the e^- or e^+. `VECT` is the new direction computed in `GTELEC` before entering in `GSYNC`.

where p is the momentum of the particle in GeV. B is in tesla and ρ is in meters.

Knowing the step length, the energy of the electron and the curvature of the particle track in the magnetic field, the number of photons in a step δx can be sampled from a Poissonian distribution around the mean value

$$\overline{n_{\delta x}} = 0.01053 \frac{\gamma}{\rho} \delta x \tag{6}$$

Now, the energies of $n_{\delta x}$ photons have to be determined. The energy distribution in a step follows the distribution [81]

$$f(\varepsilon) \; = \; \frac{dN}{d\varepsilon} \; \propto \; \int_{\varepsilon/\varepsilon_c}^{\infty} K_{5/3}(x)\, dx \tag{7}$$

The energy can be sampled from this by inverse transform method:

$$\begin{aligned} N &= F(\varepsilon) = \int f(\varepsilon)\, d\varepsilon && (8) \\ \varepsilon &= F^{-1}(N) && (9) \end{aligned}$$

The double integral is not analytically solved, and the sampling is done from tabulated values of numerically computes $F(\varepsilon)$.

Two methods have been implemented. If the flag `ISYNC` is set to 1, the photons are emitted at the end of the step along the current direction. If `ISYNC` is set to 3, the photons are emitted randomly along the tangent the real trajectory of the particle. In the case when `ISYNC = 3`, the magnetic field tracking routines are called for each photon, and therefore this option is considrably slower than `ISYNC = 1`.

Origin : G.N.Patrick
Revision :
Documentation :

Submitted: 30.03.82
Revised: 16.12.93

Simulation of particle decays in flight

1 Subroutines

`CALL` **GDECAY**

`GDECAY` is the control routine for the simulation of particle decays in flight. For a given parent particle it selects from a list a two- or three-body decay mode using the known branching ratios and calls the routines needed to generate the vertex and secondary tracks. It used the following input and output:

`input:` via common blocks `/GCTRAK/` and `/GCKINE/`
`output:` via common block `/GCKING/`

`GDECAY` is called by the tracking routines. It calls the subroutines `GDECA2` for two-body decay, `GDECA3` for three-body decay, `GLOREN` for Lorentz transformation and `GDROT` for rotation. For the documentation of `GLOREN` and `GDROT`, see `[PHYS410]`.

`CALL` **GDECA2** `(XM0,XM1,XM2,PCM)`

`XM0` (`REAL`) mass of the parent particle
`XM1` (`REAL`) mass of the first decay product
`XM2` (`REAL`) mass of the second decay product
`PCM(3,4)` (`REAL`) array containing the four-vectors of the decay products

`GDECA2` simulates the two-body decay with isotropic angular distribution in the center-of-mass system. It is called from `GDECAY`.

`CALL` **GDECA3** `(XM0,XM1,XM2,XM3,PCM)`

`XM0` (`REAL`) mass of the parent particle
`XM1` (`REAL`) mass of the first decay product
`XM2` (`REAL`) mass of the second decay product
`XM3` (`REAL`) mass of the third decay product
`PCM(3,4)` (`REAL`) array containing the four-vectors of the decay products

`GDECA3` simulates the three-body decay with isotropic angular distribution in the center-of-mass system. It is called from `GDECAY`.

2 Method

- Upon entry to `GDECAY` a binary search is made in a list of parent particles. This list is stored in the `JPART` structure and currently contains the particles defined in `GPART [CONS300]`. If the current particle cannot be found in the list, 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 `[CONS310]`).
- A decay channel is selected according to the branching ratios. If the sum of the branching ratios for a particle is not equal 100%, it is possible that *no* decay is selected.

- Depending on whether the two- or three-body decay is selected, either GDECA2 or GDECA3 is called to generate the four-momenta of the decay products with isotropic angular distribution in the center-of-mass system.
- The momentum vectors of the decay products are transformed into the laboratory system and rotated back into the GEANT coordinate frame. The kinematics of the products is stored in the common /GCKING/.
- When a particle decays and no branching ratio is defined, then GDECAY calls the user routine GUDCAY.

Origin : M.Hansroul
Revision :
Documentation :

Submitted: 01.09.76
Revised: 16.12.93

Rotations and Lorentz transformation

1 Subroutines

```
CALL GLOREN (BETA,PA,PB*)
```

GLOREN transforms the momentum and the energy from one Lorentz frame (A) to another (B). It uses the following input and output:

BETA(4) (REAL) velocity components in light units of frame B in frame A, BETA(1...3) $= \vec{\beta}$, BETA(4)$= (1-\beta^2)^{-1/2} = \gamma$;
PA(4) (REAL) momentum components in the frame (A);
PB(4) (REAL) momentum components in the frame (B).

GLOREN is called from the routine GDECAY. The momentum $\vec{p}_A$ and energy E_A in reference frame A are transformed into the momentum $\vec{p}_B$ and energy E_B in reference frame B which translates with velocity $\vec{\beta}$ with respect to A:

$$
\begin{aligned}
E_B &= \gamma(E_A - \vec{\beta}\cdot\vec{p}_A) \\
\vec{p}_B &= \vec{p}_A + \gamma\vec{\beta}\left(\frac{\gamma\vec{\beta}\cdot\vec{p}_A}{\gamma+1} - E_A\right)
\end{aligned}
$$

```
CALL GDROT (P*,COSTH,SINTH,COSPH,SINPH)
```

GDROT rotates a vector from one reference system to another.

P(3) (REAL) vector to rotate, overwritten on output;
COSTH (REAL) cosine of the polar angle;
SINTH (REAL) sine of the polar angle;
COSPH (REAL) cosine of the azimuthal angle;
SINPH (REAL) sine of the azimuthal angle;

GDROT is called from several routines to rotate a particle from in the center-of-mass system to the GEANT (laboratory) system. The following rotation matrix is used:

$$
\begin{pmatrix} \cos\theta\cos\phi & -\sin\phi & \sin\theta\cos\phi \\ \cos\theta\sin\phi & \cos\phi & \sin\theta\sin\phi \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{R_\phi} \underbrace{\begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}}_{R_\theta}
$$

R_θ is a counterclockwise rotation around axis y by an angle θ, and R_ϕ is a counterclockwise rotation around axis z' (rotated by an angle θ from the initial position) by an angle ϕ.

```
CALL GFANG (P,COSTH*,SINTH*,COSPH*,SINPH*,ROTATE*)
```

P(3) (REAL) input direction;
COSTH (REAL) cosine of the polar angle;

SINTH (REAL) sine of the polar angle;
COSPH (REAL) cosine of the azimuthal angle;
SINPH (REAL) sine of the azimuthal angle;
ROTATE (LOGICAL) rotation flag, the rotation matrix is the unit matrix if ROTATE=.FALSE., i.e. no rotation is needed;

Find the sine and cosine of the polar and azimuthal angle of the direction defined by the vector P. If P is along the z axis, ROTATE=.FALSE..

```
CALL GVROT (DCOSIN,PART*)
```

PART(3) (REAL) direction to be rotated, overwritten on output;
DCOSIN(3) (REAL) direction of the new z axis;

Given the direction PART in a system of reference O, it returns the same direction in the system of reference where the z axis of O is along DCOSIN. This routine is very useful when the θ angle of a secondary particle is sampled in a reference frame where the parent particle moves along the z axis. Then the direction of the secondary product in the MRS is returned by GVROT if DCOSIN is the direction of the parent particle.

A call to GVROT is equivalent to the following code:

```
DIMENSION PART(3), DCOSIN(3)
LOGICAL ROTATE
.
.
.
CALL GFANG(DCOSIN,COSTH,SINTH,COSPH,SINPH,ROTATE)
IF(ROTATE) CALL GDROT(PART,COSTH,SINTH,COSPH,SINPH)
```

with the advantage that inside GVROT all calculations are performed in double precision.

Origin : G.N.Patrick, L.Urbán, D.Ward
Submitted: 12.03.82
Revision :
Revised: 16.12.93
Documentation :

Ionisation processes for muons and protons

1 Subroutines

CALL **GDRELA**

GDRELA initialises the ionisation energy-loss tables for different materials and particles. The energy binning is in the array ELOW (/GCMULO/) initialised in the routine GPHYSI. The tables are filled with the quantity dE/dx in GeV cm^{-1}, for elements, mixtures and compounds. A temporary table is also filled, containing additional points for integration, to obtain the range-energy table (see [PHYS010]).

GDRELP computes the ionisation energy loss for protons, and the values are stored in the energy loss table. This value is used for other charged hadrons with scaled energy bin.

GDRELM computes the ionisation energy loss for muons, and sums it in the muon table with the energy loss due to the other processes (bremsstrahlung, pair production and nuclear interactions).

The following pointers are used:

JMA = LQ(JMATE-I)	Ith material bank;
JEL1 = LQ(JMA-1)	dE/dx for e^+/e^-(see [PHYS330]);
JEL2 = LQ(JMA-2)	dE/dx for μ^+/μ^-;
JEL3 = LQ(JMA-3)	dE/dx for protons (used also for other charged particles).

GDRELA is called during initialisation by GPHYSI.

CALL **GDRELP** (A,Z,DENS,T,DEDX*)

GDRELP computes the ionisation energy loss for protons. It is called by the routine GDRELA. See GDRELX below for the meaning of the parameters.

CALL **GDRELM** (A,Z,DENS,T,DEDX*)

GDRELM computes the energy loss due to ionisation for muons. It is called by the routine GDRELA. See GDRELX below for the meaning of the parameters.

CALL **GDRELX** (A,Z,DENS,T,HMASS,DEDX*)

A	(REAL) mass number of the material;
Z	(REAL) atomic number of the material;
DENS	(REAL) density of the material in g cm^{-3}; it is only used to calculate the density correction in case of high values of T;
T	(REAL) kinetic energy of the particle in GeV;
HMASS	(REAL) mass of the particle;
DEDX	(REAL) energy loss of the particle in GeV cm^2 g^{-1}.

This routine calculates the ionisation energy loss for a muon or a proton. It is called by GDRELM and GDRELP.

CALL **GDRSGA**

CDRSGA calculates the total cross-section for the production of δ-rays by electrons (Möller scattering), positrons (Bhabha scattering) and muons. For hadrons, this value is calculated at tracking time. The mean free path as a function of the energy is stored for every medium. The following pointers are used:

`JMA = LQ(JMATE-I)`	pointer to the I^{th} material;
`JDRAY = LQ(JMA-11)`	δ-ray cross-section for electrons;
`JDRAY+NEK1`	δ-ray cross-section for positrons;
`JDRAY+2*NEK1`	δ-ray cross-section for muons.

The routine is called during initialisation by `GPHYSI`.

2 Method

In `GEANT`, δ-rays are generated only above a threshold energy T_{cut} which corresponds to the variable `DCUTE` for electrons and positrons, `DCUTM` for all other charged particles. The total cross-section for the production of a δ-ray electron of kinetic energy $T > T_{cut}$ =`DCUTE, DCUTM` by a particle of kinetic energy E is:

$$\sigma(E, T_{cut}) = \int_{T_{cut}}^{T_{max}} \frac{d\sigma(E,T)}{dT} dT \tag{1}$$

where T_{max} is the maximum energy transferable to the free electron:

$$T_{max} = \frac{2m(\gamma^2 - 1)}{1 + 2\gamma\frac{m}{M} + \left(\frac{m}{M}\right)^2} \tag{2}$$

The energy lost in ionising collisions producing δ-rays below T_{cut} are included in the continuous energy loss. The mean value of the energy lost due to these *soft* collisions is:

$$E_{loss}(E, T_{cut}) = \int_{0}^{T_{cut}} T \frac{d\sigma(E,T)}{dT} dT \tag{3}$$

where m is the electron mass and M is the mass of the incident particle.

In this chapter, the method of calculation of the continuous energy loss and the total cross-section are explained. The generation of δ-rays is explained in chapter `[PHYS331]`.

2.1 Continuous energy loss

The integration of (3) leads to the Bethe-Block stopping power or to the restricted energy loss formula [55]:

$$\frac{1}{\rho}\left(\frac{dE}{dx}\right) = \begin{cases} D\frac{ZZ_{inc}^2}{A\beta^2}\left[\ln\left(\frac{2m_e\beta^2\gamma^2 T_{max}}{I^2}\right) - 2\beta^2 - \delta - \frac{2C_e}{Z}\right] & \text{if} \quad T_{cut} \geq T_{max} \\ D\frac{ZZ_{inc}^2}{A\beta^2}\left[\ln\left(\frac{2m_e\beta^2\gamma^2 T_c}{I^2}\right) - \beta^2\left(1 + \frac{T_c}{T_{max}}\right) - \delta - \frac{2C_e}{Z}\right] & \text{if} \quad T_{cut} < T_{max} \end{cases} \tag{4}$$

where,

D	$2\pi N_A r_e^2 mc^2 = 0.000153537$ GeV cm^2 g^{-1}	Z	atomic number of medium
A	mass number	ρ	density of medium
T_c	$\min(T_{cut}, T_{max})$	m_e	electron mass

and I is the average ionisation potential of the atom in question. There exists a variety of phenomenological approximations for this. In former versions of `GEANT` the formula quoted by [82] was used ($I = 16\ Z^{0.9}$ eV). At present the values recommended by [83] are used. The ionisation potential I only enters into the logarithmic term of the energy loss formula, but it has been verified that the new parameterisation gives better accuracy especially in the case of high Z. It should be noted that this is not the value of I which is stored in the material data structure by `GPROBI`, which is still calculated as $I = 16\ Z^{0.9}$ eV.

Note: the ionisation potential I must not be changed blindly, hoping that the most up-to-date values give the better results. The value of I is closely connected to the shell correction term (see later).

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 [84, 85] is used:

$$\delta = \begin{cases} 0 & \text{if} \quad X < X_0 \\ 4.606X + C + a(X_1 - X)^m & \text{if} \quad X_0 \leq X < X_1 \\ 4.606X + C & \text{if} \quad X \geq X_1 \end{cases} \tag{5}$$

where the medium-dependent constants are calculated as follows:

$$\begin{array}{lcllcl} X & = & \log_{10}(\gamma\beta) = \ln(\gamma^2\beta^2)/4.606 & \nu_p & = & \sqrt{\dfrac{N_{el}e^2}{\pi m}} \ \text{s}^{-1} \text{ plasma frequency} \\ N_{el} & = & \dfrac{\rho Z N_{Av}}{A} \ \text{electrons cm}^{-3} & C & = & -2\ln\left(\dfrac{I}{h\nu_p}\right) - 1 \\ a & = & \dfrac{4.606(X_a - X_0)}{(X_1 - X_0)^m} & 4.606\,X_a & = & -C \end{array}$$

For condensed media we have:

$$I < 100\,\text{eV} \quad \begin{cases} X_0 & = & \begin{cases} 0.2 & \text{if} \quad -C \leq 3.681 \\ -0.326C - 1.0 & \text{if} \quad -C > 3.681 \end{cases} \\ X_1 & = & 2.0 \\ m & = & 3.0 \end{cases}$$

$$I \geq 100\,\text{eV} \quad \begin{cases} X_0 & = & \begin{cases} 0.2 & \text{if} \quad -C \leq 5.215 \\ -0.326C - 1.5 & \text{if} \quad -C > 5.215 \end{cases} \\ X_1 & = & 3.0 \\ m & = & 3.0 \end{cases}$$

and in the case of gaseous media $m = 3$ and:

$$\begin{array}{llll} X_0 = 1.6 & X_1 = 4 & \text{for} & C \leq 9.5 \\ X_0 = 1.7 & X_1 = 4 & \text{for} & 9.5 < C \leq 10 \\ X_0 = 1.8 & X_1 = 4 & \text{for} & 10 < C \leq 10.5 \\ X_0 = 1.9 & X_1 = 4 & \text{for} & 10.5 < C \leq 11 \\ X_0 = 2. & X_1 = 4 & \text{for} & 11 < C \leq 12.25 \\ X_0 = 2. & X_1 = 5 & \text{for} & 12.25 < C \leq 13.804 \\ X_0 = 0.326C - 2.5 & X_1 = 5 & \text{for} & 13.804 < C \end{array}$$

C_e/Z is a so-called *shell correction term* which accounts for the fact that, at low energies for light elements and at all energies for heavy ones, the probability of collision with the electrons of the inner atomic shells (K, L, etc.) is negligible. The semi-empirical formula used in GEANT, applicable to all materials, is due to Barkas [86]:

$$\begin{array}{rcl} C_e(I, \eta) & = & (0.42237\eta^{-2} + 0.0304\eta^{-4} - 0.00038\eta^{-6})10^{-6}I^2 \\ & + & (3.858\eta^{-2} - 0.1668\eta^{-4} + 0.00158\eta^{-6})10^{-9}I^3 \\ \eta & = & \gamma\beta \end{array}$$

C_e is a dimensionless constant, but as I in the original article was expressed in eV and in GEANT it is expressed in GeV, the exponent of ten in the I^2-term is $10^{-6+2\times 9} = 10^{12}$, and that of the I^3-term is $10^{-9+3\times 9} = 10^{18}$.)

$$\frac{dE}{dx}^{calculated} - \frac{dE}{dx}^{measured}$$

experiment	A [89]	B [90]	C [91]
projectile	p	p	p
T (MeV)	0.03-0.6	0.4-1	55,65,73
material	H_2, He, N_2, O_2, Ne, Ar, Xe	Cu, Ge, Sn, Pb	Al, Ti, Cu, Sn, Pb
exp err (%)	~ 3	~ 2.5	~ 0.7
tot N° of pts	121	52	15
N° of pts			
with $\lvert\Delta\rvert < 1\sigma$	94	50	8
$\lvert\Delta\rvert < 2\sigma$	114	52	13
$\lvert\Delta\rvert < 3\sigma$	119	52	14
$\lvert\Delta\rvert < 4\sigma$	121	52	15

Table 1.7: Test of GDRELP with low energy protons.

This formula breaks down at low energies, and it only applies for values of $\eta > 0.13$ (i.e. $T > 7.9$ MeV for a proton). For $\eta \leq 0.13$ the shell correction term is calculated as:

$$C_e(I,\eta)\Big|_{\eta\leq 0.13} = C_e(I,\eta = 0.13)\frac{\ln\left(\frac{T}{T_{2l}}\right)}{\ln\left(\frac{7.9\,10^{-3}\,GeV}{T_{2l}}\right)}$$

i.e. the correction is *switched off* logarithmically from $T = 7.9$ MeV to $T = T_{2l} = 2$ MeV. GDRELX has been tested for protons against energy loss tables [87, 88] for various materials in the energy range 50 MeV-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

The mean energy loss can be described by the Bethe-Bloch formula (4) only if the projectile velocity is larger than that of orbital electrons. In the low-energy region where this is not verified, a different kind of parameterisation has been chosen [83]:

$$\frac{1}{\rho}\frac{dE}{dx} = \begin{cases} I: & C_1\tau^{\frac{1}{2}} & \text{for} \quad 0 \leq T \leq T_{1l} = 10\,\text{keV} \\ II: & \dfrac{S_L \times S_H}{S_L - S_H} & \text{for} \quad T_{1l} < T \leq T_{2l} = 2\,\text{MeV} \\ III: & \{\text{Bethe-Bloch}\}\left(1 + \dfrac{\nu}{T}\right) & \text{for} \quad T_{2l} < T \end{cases}$$

where

$$\begin{aligned} S_L &= C_2\tau^{0.45} & \qquad S_H &= \frac{C_3}{\tau}\ln\left[1 + \frac{C_4}{\tau} + C_5\tau\right] \\ \tau &= \frac{T}{M_p} & \qquad M_p &= \text{proton mass} \end{aligned}$$

$1 \text{ keV} \leq T \leq 100$ MeV, 31 pts per element [83]			
Z	element	mean (r.m.s.) deviation in %	max (r.m.s.) deviation in %
1	H_2	0.4	1.1
6	O	0.5	1.6
13	Al	0.6	2.1
29	Cu	0.7	2.0
82	Pb	0.7	2.3

Table 1.8: Test of GDRELP against stopping power tables. Stated accuracy of the tables is $\sim 5\%$ for $T > 0.5$ MeV and $\sim 10\%$ for $T < 0.5$ MeV

$$\Delta = \frac{\frac{dE}{dx}^{min}_{calc} - \frac{dE}{dx}^{min}_{table}}{\frac{dE}{dx}^{min}_{table}} (\%)$$

Data from [92]

material	Δ (%)	material	Δ (%)	material	Δ (%)
H_2	-0.4	Na	-0.4	Sn	1.7
D_2	-0.3	Al	-0.6	Xe	1.0
He	0.0	Si	0.5	W	1.1
Li	3.1	Ar	0.4	Pt	0.3
Be	-0.2	Ti	-1.4	Pb	1.5
C	-0.1	Fe	-0.4	U	1.4
N_2	-0.2	Cu	-0.9		
O_2	-1.2	Ge	-0.8		

Table 1.9: The minimum stopping power calculated from the formula used is compared with values from the tables.

The formula used in the region $T > T_{2l}$ ensures the continuity of the energy loss function at $T = T_{2l}$ when the Bethe-Bloch formula and the parameterisation meet. The parameter ν is chosen in such a way that:

$$\frac{1}{\rho}\frac{dE}{dx}^{II}\bigg|_{T=T_{2l}} = \frac{1}{\rho}\frac{dE}{dx}^{III}\bigg|_{T=T_{2l}}$$

The routine GDRELP calculates the stopping power or restricted energy loss only for $T > T_{2l}$; below this kinetic energy it gives the stopping power (i.e. total energy loss) irrespectively of the value of T_{cut}. This approximation does not introduce a serious source of error since, in the case of a proton, at $T = T_{2l}$ the maximum energy transferable to the atomic electron is $T_{max} \approx 4$ keV, and the restricted loss should be calculated only if $T_{cut} < T_{max}$.

GDRELP has been tested against experimental data and energy loss tables. Some of the test results are summarised in tables 2.1, 2.1 and 2.1.

The energy lost due to the soft δ-rays is tabulated during initialisation as a function of the medium and of the energy by routine GDRELA

The tables are filled with the quantity dE/dx in GeV cm^{-1} (formula (4) above). For a molecule or a mixture the following formula is used:

$$\frac{dE}{dx} = \rho \sum_i p_i \left(\frac{dE}{dX}\right)_i \tag{6}$$

where x is in cm, X in g cm^{-2} and p_i is the proportion by weight of the Ith element.

The energy loss of all charged particles other than electrons, positrons and muons is obtained from that of protons by calculating the kinetic energy of a proton with the same β, and using this value to interpolate the tables:

$$T_{proton} = \frac{M_p}{M}T \tag{7}$$

2.2 Total cross-section

The integration of formula (1) gives the total cross-section :

$$\sigma(Z,E,T_{cut}) = \begin{cases} 2\pi r_0^2 m \dfrac{Z}{\beta^2} \dfrac{1 - Y + \beta^2 Y \ln Y}{T_{cut}} & \text{for spin-0 particles} \\ 2\pi r_0^2 m \dfrac{Z}{\beta^2} \left(\dfrac{1 - Y + \beta^2 Y \ln Y}{T_{cut}} + \dfrac{T_{max} - T_{cut}}{2E^2} \right) & \text{for spin-}\dfrac{1}{2}\text{ particles} \end{cases} \tag{8}$$

The mean free path is tabulated during initialisation as a function of the medium and of the energy by the routine `GDRSGA` *for leptons only*.

The cross-section (8) is strongly dependent on the mass of the incident particle, and cannot be tabulated in a general way for any charged hadrons. Therefore, for such particles, the cross-section is computed at tracking time in the routine `GTHADR`.

Origin : L.Urbán
Revision :
Documentation :

Submitted: 12.03.82
Revised: 16.12.93

Ionisation processes for heavy ions

1 Subroutines

For a description of the subroutines used to calculate the energy loss tables for protons, see [PHYS430].

CALL **GTHION**

GTHION transports heavy ions ($A, Z > 1$, particle type 8). For the moment the only discrete processes activated are δ-ray and Čerenkov photon generation.

2 Method

The mean energy loss of the heavy ions can be expressed as:

$$\Delta E = Q^2 S_p \left(\frac{T}{A}\right) \rho t \tag{1}$$

where

Q	charge of the ion;
$S_p\left(\frac{T}{A}\right)$	energy loss by the a proton with kinetic energy $T_p = T/A$;
T, A	kinetic energy and mass number of the ion;
ρ	density of the medium;
t	step length

This formula is used to calculate the ionisation loss for all the charged hadrons.

In the case of ions, the charge of the ion makes the problem more difficult. Electron-exchange processes with the atoms change the charge as the ion traverses the medium. The main features of this process can be summarised as follows:

- Q loses the dependence on the initial ion charge after a very short initial step, usually $\ll 1\mu$ for condensed media;
- Q does depend on the velocity of the ion;
- even in the case of constant velocity, Q fluctuates around a mean value;
- Q features a small dependence on the medium, especially for small values of T.

If we want to calculate the energy loss of the ion, we need a good parametrisation for the quantity $Q = Q_{eff}$. We give here a relatively simple parametrisation [83, 93]:

$$\begin{aligned} Q_{eff} &= Z_I \left[1 - \left(1.034 - 0.1777 e^{-0.08114\, Z_I}\right) e^{-v}\right] \\ v &= 121.4139 \frac{\beta}{Z_I^{2/3}} + 0.0378 \sin\left(190.7165 \frac{\beta}{Z_I^{2/3}}\right) \end{aligned} \tag{2}$$

where β is the velocity and Z_I the atomic number of the ion (i.e. the charge of the bare nucleus).

It can be seen that (2) neglects the (small) medium dependence of Q_{eff}. For very high energies ($\beta \to 1$) $Q_{eff} \to Z_I$. For very low energies ($T/A \sim$ few keV) the formula breaks down. Q_{eff} can even become

T/A MeV	He ions error (%)	Pb ions error (%)
10	3	1
5	4	3
3	10	4

Table 1.10: Comparison of the measured and calculated ΔE for 10 MeV/A ions in Pb

T (MeV)	417 μg cm^{-2}						110 μg cm^{-2}					
	MC (keV)				data (keV)		MC (keV)				data (keV)	
	ΔE	dif	FWHM	dif	ΔE	FWHM	ΔE	dif	FWHM	dif	ΔE	FWHM
4.88							785	8	59	-23	725	77
9.85	3600	14	138	-33	3150	206	816	8	67	-26	756	91
19.79	3090	3	142	-35	2990	218	719	3	70	-32	699	103
29.27	2660	1	141	-31	2630	204						
29.75							620	4	69	-26	598	93
39.70	2320	1	138	-28	2300	191	550	4	68	-24	528	90

Table 1.11: Comparison of the measured and calculated ΔE and FWHM for O ions in Al; errors are in percent.

negative for $T/A < 20$ keV and $Z_I > 20$. However this is not a serious source of error when calculating ΔE, since in this case the range of the ion is very small, and it can almost be said that it stops *immediately*.

The calculation of the energy loss straggling (fluctuations) differs from that of *normal* charged hadrons. For the charged hadrons the fluctuations come from the statistical nature of the projectile-atom interactions; for the ions there is another process which broadens the energy loss distribution: the fluctuation of the charge. For heavier ions this process dominates the energy loss straggling for $T/A \leq 10$ MeV.

The heavy ions are in the *Gaussian regime* (see [PHYS332]) of the collisional fluctuations even in the case of very thin absorbers. If T/A is not too high, the σ^2 of the distribution:

$$\sigma^2_{coll} = D\frac{Q^2_{eff}Z}{A}\rho\ t\ m\left[1+\frac{T_A}{M_u}+\frac{1}{2}\left(\frac{T_A}{M_u}\right)^2\right] \tag{3}$$

where

D	$0.307\frac{\text{MeV cm}^2}{\text{g}}$;
T_A	$\frac{T}{A}$;
M_u	atomic mass unit;
Z	atomic number of the medium;
A	mass number of the medium.

Analysing the experimental straggling data it is possible to find that the electron-exchange charge fluctuations can be described by a Gaussian with width:

$$\sigma^2_{ch} = D\frac{Q^2_{eff}Z}{A}\rho\ t\ m\frac{C}{2}\left(1-\frac{Q_{eff}}{Z_I}\right) \tag{4}$$

where the parameter $C \sim 2.5$ has been derived from the experimental straggling data.

If $Q_{eff} \to Z_I$, which is the case for high energy heavy ions and for few MeV/A He ions, then $\sigma^2_{ch} \to 0$.

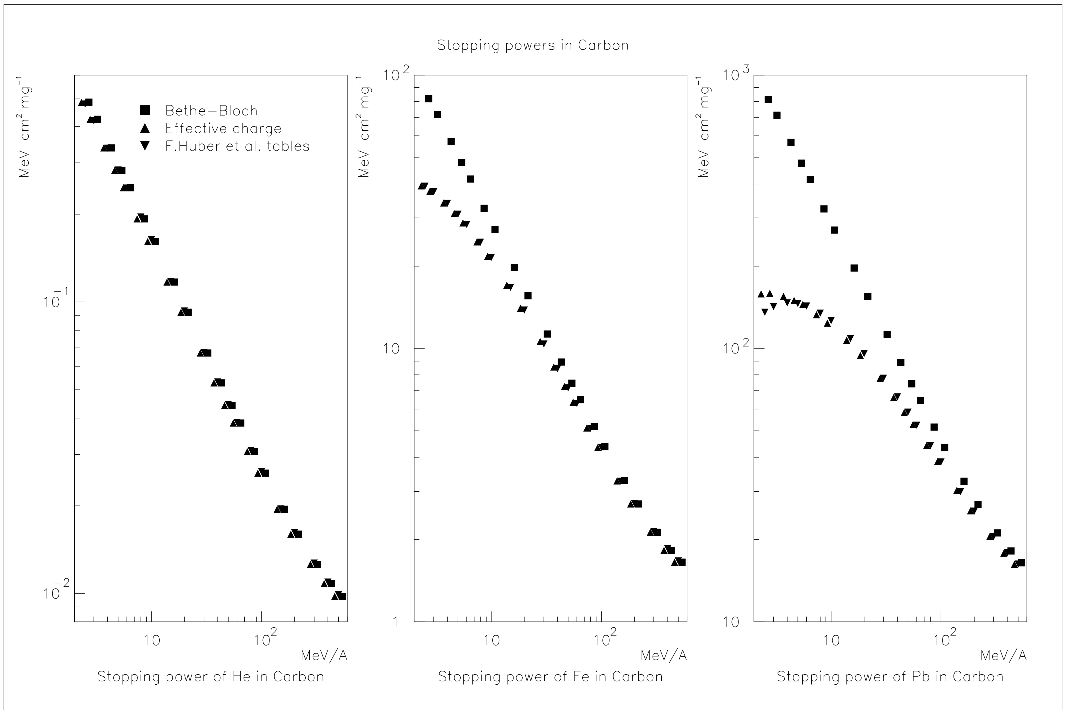


Figure 43: Stopping powers in Carbon

	Pb ions in gas (energies in MeV)								
	N_2			Ar			Xe		
	MonteCarlo		data	MonteCarlo		data	MonteCarlo		data
t (cm)	ΔE	FWHM	FWHM	ΔE	FWHM	FWHM	ΔE	FWHM	FWHM
0.2	25.9	1.15	1.1	25.4	1.30	1.4	51.1	2.26	2.5
0.4	53.6	1.62	1.5	52.4	1.84	1.9	107.0	3.22	3.3
0.8	112.0	2.25	2.0	111.0	2.59	2.6	236.0	4.51	
1.2	162.0	2.37	2.0	168.0	2.95	2.8			

Table 1.12: Comparison of the measured and calculated and FWHM for 1.4 MeV/A Pb ions in gas.

Comparing equations (3) and (4) it can be seen that for heavy ions and for $T/A \ll M_u$ $\sigma^2_{ch} > \sigma^2_{coll}$. The total energy loss fluctuation can be described by a Gaussian distribution with:

$$\sigma^2 = \sigma^2_{ch} + \sigma^2_{coll} \tag{5}$$

The mean energy loss and energy loss fluctuation calculation is performed in the routine GTHION, making use of the proton energy loss tables.

Note: The Gaussian fluctuation gives too broad a distribution for high energy in the case of thin absorbers. A correction has been introduced in GTHION which cures this discrepancy. In the absence of high energy straggling data for ions, the correction has been tuned using high energy π energy loss data, where the π has been tracked by GTHION.

3 Comparison with data

3.1 Mean energy loss and range

A test has been made against the distributions reported in [94] for He ions at 1 and 10 MeV/A and for O and Pb ions at 1, 10 and 100 MeV/A in carbon and the results were found to be correct within 5% both for

He ions in Ar (energies in keV)			
	MonteCarlo		data
t (cm)	ΔE	FWHM	FWHM
0.5	444	38	∼40
1.0	913	53	∼60
2.0	1940	78	∼80

Table 1.13: Comparison of the measured and calculated and FWHM for 1.4 MeV/A He ions in gas.

the energy loss and the range.

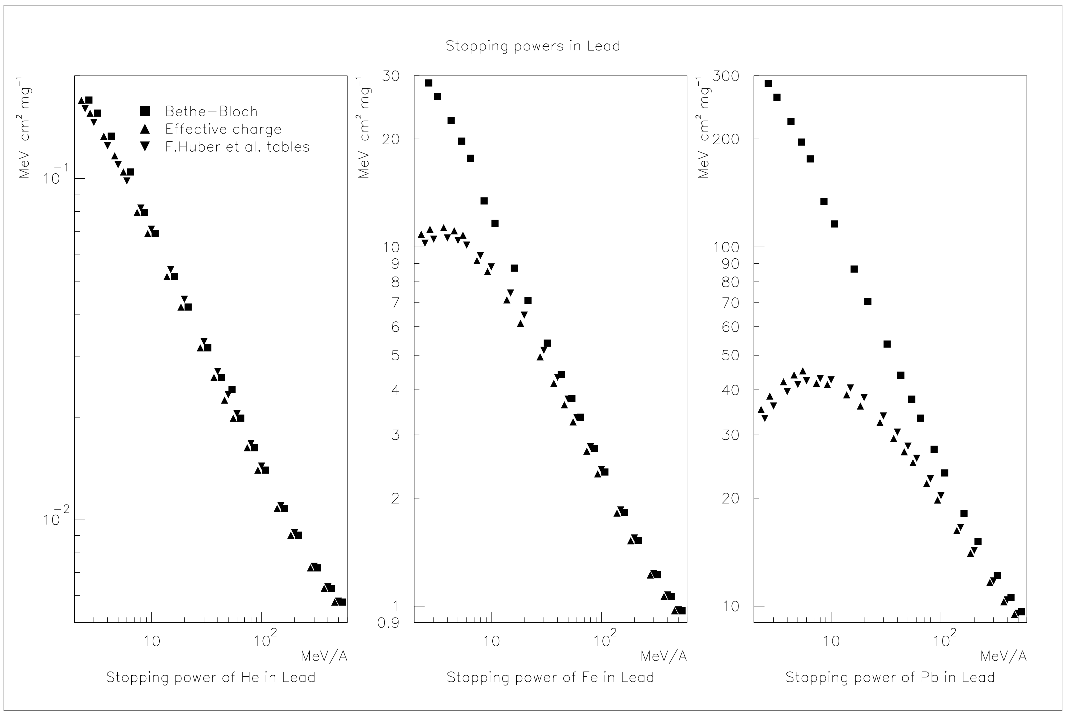


Figure 44: Stopping powers in Lead

Another comparison has been made with the tables of [95] for 10 MeV/A ions in lead, with the results shown in table 2. Another comparison with the same tables is shown in Figs. 43 and 44.

A comparison with the tables in [83, 93] for T/A $<$ 1 MeV/A gives an error on $(dE/dx) \leq 10 - 20\%$.

3.2 Comparison of FWHM

A comparison has been made with low energy data ($0.3 \leq T/A \leq 2.5$ MeV/A) in very thin absorbers (1,7 μm, 0.4 μm). Data are taken from the tables in [96] for energy loss of O ions in Al, and the results are reported in table 2. The quoted experimental error for these data is 1% statistical + 5% systematic for 417 μg cm^{-2} data and 2% statistical + 10% systematic for 110 μg cm^{-2} data.

Plots of energy deposition distributions of 1.4 MeV/A lead and helium ions in gases can be found in [97]. From these we have measured the FWHM and compared them with the ones given by the MonteCarlo. The results can be seen in table 2 and 3.1. It should be noted that the absorber thickness of, for example, Ar compares with that of the preceding case (0.2 cm Ar = 356 μg cm^{-2} and 1.2 cm Ar = 2136 μg cm^{-2}).

Similar comparison has been made for C ions of energies 17.1 and 39.4 MeV in isobuthane [98] and the agreement between data and `GEANT` for FWHM is within 30%.

Origin : L.Urbán
Revision :
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Total cross-section and energy loss for bremsstrahlung by Muons

1 Subroutines

`CALL` **GBRELA**

`GBRELA` fills the tables for continuous energy loss of electrons, positrons and muons due to bremsstrahlung during initialisation time. The energy binning is determined by the array `ELOW` (common `/CGMULO/`) in the routine `GPHYSI`. In the tables, the dE/dx due to bremsstrahlung is summed with that due to the ionisation. For energy loss of muons, `GBRELA` calls the function `GDRELM`. The following pointers are used:

`JMA = LQ(JMATE-I)`	pointer to the Ith material;
`JEL1 = LQ(JMA-1)`	pointer to dE/dx for electrons;
`JEL1+NEK1`	pointer to dE/dx for positrons;
`JEL2 = LQ(JMA-2)`	pointer to dE/dx for electrons.

`GBRELA` is called by `GPHYSI`.

`VALUE` = **GBRELM** `(Z,T,BCUT)`

`Z`	(`REAL`) atomic number of the material;
`T`	(`REAL`) kinetic energy of the muon;
`BCUT`	(`REAL`) *soft* bremsstrahlung cut.

`GBRELM` calculates the muon energy loss due to bremsstrahlung of photons with energy below `BCUT`, (variable `BCUTM` in the common `/GCCUTS/`). Above this value, the bremsstrahlung process is simulated explicitly (see `[PHYS441]`) and the energy lost by the muons is not included in the tables. In the tables, the dE/dx due to bremsstrahlung is summed with the energy lost coming from ionisation, pair production and nuclear interaction. `GBRELM` is called by `GBRELA`.

`CALL` **GBRSGA**

`GBRSGA` calculates the total cross-section for bremsstrahlung. It tabulates the mean free path in cm as a function of the medium and the energy. The energy binning is determined by the array `ELOW` (common `/CGMULO/`). The following pointers are used:

`JMA = LQ(JMATE-I)`	pointer to the Ith material;
`JBREM = LQ(JMA-9)`	pointer to bremsstrahlung cross-sections;
`JBREM`	pointer for e$^-$;
`JBREM+NEK1`	pointer for e$^+$;
`JBREM+2*NEK1`	pointer for μ^+/μ^-.

`GBRSGA` is called during initialisation by `GPHYSI`.

`VALUE` = **GBRSGM** `(Z,T,BCUT)`

`Z`	(`REAL`) atomic number of the materian;
`T`	(`REAL`) kinetic energy of the muon;
`BCUT`	(`REAL`) *soft* bremsstrahlung cut.

`GBRSGM` calculates the total bremsstrahlung cross-section for muons when the emitted photon has an energy greater than `BCUT`, (variable `BCUTM` in the common `/GCCUTS/`). It is called by `GBRSGA`.

2 Method

The mean value of the energy lost by the muon due to bremsstrahlung of photons of energy $< k_c =$ `BCUTM` is:

$$E^{brem}_{loss}(Z,T,k_c) = \int_0^{k_c} k \frac{d\sigma(Z,T,k)}{dk} dk \tag{1}$$

and the total cross-section for the emission of a photon of energy $> k_c$ is

$$\sigma(Z,T,k_c) = \int_{k_c}^{T} \frac{d\sigma(Z,T,k)}{dk} dk \tag{2}$$

Accurate cross-section formula for the high energy ($T \geq 1$ GeV) muon bremsstrahlung can be found in [99].

2.1 Parameterisation of energy loss and total cross-section

The cross-sections from [99] have been used to calculate *data* points for (1) and (2). These have been parameterised as:

$$\begin{aligned} \sigma(Z,T,k_c) &= Z[Z+\xi_\sigma(1+\gamma\ln Z)]\left[\ln\frac{k_{max}}{k_c}\right]^\alpha F_\sigma(Z,X,Y) \quad \text{in barn} \\ E^{brem}_{loss}(Z,T,k_c) &= Z[Z+\xi_l(1+\delta\ln Z)]\left[\ln\frac{k_c}{E}\right]^\beta F_l(Z,X,Y) \quad \text{in GeV barn} \end{aligned}$$

$$\begin{aligned} k_{max} &= E - 0.75\sqrt{e}m_\mu Z^{1/3} & X &= \ln\frac{E}{m_\mu} \\ Y &= \ln\frac{k_c}{m_\mu} & E &= T + m_\mu \end{aligned}$$

where k_{max} is the maximum possible value of the photon energy. The functions $F_i(Z,X,Y)$ ($i = \sigma, l$) are polynomials:

$$\begin{aligned} F_i(Z,X,Y) &= (C_1 + C_2X + \cdots + C_6X^5) + (C_7 + C_8X + \cdots + C_{12}X^5)Y \\ &+ \cdots + (C_{31} + C_{32}X + \cdots + C_{36}X^5)Y^5 \\ &+ Z[(C_{37} + C_{38}X + \cdots + C_{40}X^3) + (C_{41} + C_{42}X + \cdots + C_{44}X^3)Y \\ &+ \cdots + (C_{48} + C_{49}X + \cdots + C_{52}X^3)Y^3)] \end{aligned} \tag{3}$$

A least-squares fit has been performed on more than 2000 points for $Z = 1, 6, 13, 26, 50, 82, 92$ and 1 GeV $\leq T \leq 10$ TeV and 10 keV $\leq k_c \leq T$. The resulting values of $\xi_{\sigma,l}$, γ, α ,C_i, ξ_l, δ and β can be found in the `DATA` statements within the functions `GBRSGM` and `GBRELM` which compute formulae (1) and (2) respectively. The accuracy of the fit can be estimated as:

$$\frac{\Delta\sigma}{\sigma} = \begin{cases} 10-12\% & \text{if } T \leq 5\,\text{GeV} \\ \leq 4\% & \text{if } T > 5\,\text{GeV} \end{cases}$$

$$\frac{\Delta E^{brem}_{loss}}{E^{brem}_{loss}} = \begin{cases} 10\% & \text{if } T \leq 5\,\text{GeV} \\ \leq 3\% & \text{if } T > 5\,\text{GeV} \end{cases}$$

The contribution of the bremsstrahlung to the total energy loss of the muons is less than 1% for $T \leq 5$ GeV. When $k_c \geq k_{max}$, a parameterisation different from (3) can be used for the total muon energy loss due to bremsstrahlung:

$$\begin{aligned} E^{brem}_{loss}(Z,T) &= E^{brem}_{loss}(Z,T,k=k_{max}) \\ &= Z(Z+1)k_{max}[d_1 + (d_2X + d_3Y) + (d_4X^2 + d_5XY + d_6Y^2) \\ &+ \cdots + (d_{22}X^6 + d_{23}X^5Y + \cdots + d_{28}Y^6)] \end{aligned} \tag{4}$$

where $Y = Z^{1/3}$. The accuracy of the formula (4) for $1 \leq Z \leq 100$ is rather good:

$$\frac{\Delta E_{loss}^{brem}}{E_{loss}^{brem}} = \begin{cases} \leq 1.5\% & \text{if} \quad T > 1 \text{ GeV} \\ \leq 1\% & \text{if} \quad T > 5 \text{ GeV} \end{cases}$$

Origin : L.Urbán
Revision :
Documentation :

Submitted: 26.10. 84
Revised: 11.10.93

Simulation of discrete bremsstrahlung by muons

1 Subroutines

```
CALL GBREMM
```

`GBREMM` generates a photon from bremsstrahlung of a highly energetic muon as a discrete process. For the angular distribution of the photon, is calls the function `GBTETH`.

`Input:` via common `/GCTRAK/`;
`Output:` via common `/GCKING/`.

`GBREMM` is called from the tracking routine `GTMUON` when the muon reaches its radiation point during the tracking stage of `GEANT`.

```
VALUE = GBTETH (ENER,PARTM,EFRAC)
```

`ENER` (`REAL`) kinetic energy of the muon;
`PARTM` (`REAL`) mass of the radiating particle (m_μ in this case);
`EFRAC` (`REAL`) ratio of the energy of the radiated photon to the energy of the muon.

`GBTETH` calculates the angular distribution of the e^-e^+pair in photon pair production and of the photon in bremsstrahlung. In case of bremsstrahlung it gives the scaled angle ($\eta = E\theta m^{-1}$) of the photon.

2 Method

The differential cross-section for the emission of a photon of energy k by a muon of energy E is [99, 100]:

$$\frac{d\sigma}{dv} = \frac{N}{v}\left(\frac{4}{3} - \frac{4}{3}v + v^2\right)\Phi(\delta) \tag{1}$$

where

$$N = \text{normalisation factor} \qquad v = \frac{k}{E}$$
$$\delta = \frac{m_\mu^2}{2E}\frac{v}{1-v} \qquad e = 2.718\ldots$$
$$\Phi(\delta) = \begin{cases} \ln\left(\frac{189 m_\mu}{m_e Z^{1/3}}\right) - \ln\left(\frac{189\sqrt{e}}{m_e Z^{1/3}}\delta + 1\right) & \text{if} \quad Z \leq 10 \\ \ln\left(\frac{189 m_\mu}{m_e Z^{1/3}}\right) - \ln\left(\frac{189\sqrt{e}}{m_e Z^{1/3}}\delta + 1\right) + \ln\left(\frac{2}{3}Z^{-1/3}\right) & \text{if} \quad Z > 10 \end{cases}$$
$$v_c = \frac{k_c}{E} \leq v \leq \left(1 - 0.75\sqrt{e}\frac{m_\mu}{E}Z^{1/3}\right) = v_{max}$$

Therefore, the differential cross-section can be written as

$$\frac{d\sigma}{dv} = f(v)g(v) \tag{2}$$

with

$$f(v) = \left[v \ln \left(\frac{v_{max}}{v_c} \right) \right]^{-1}$$

$$g(v) = \frac{1}{\Phi(0)} \left(1 - v + \frac{3}{4} v^2 \right) \Phi(\delta)$$

We can sample the photon energy in the following way (r_1, r_2 uniformly distributed random numbers in $]0, 1[$):

1. sample v:

$$v = v_c \left(\frac{v_{max}}{v_c} \right)^{r_1}$$

2. compute the rejection function $g(v)$ and:
   (a) if $r_2 > g(v)$ go back to step 1
   (b) if $r_2 \leq g(v)$, accept v and $k = vE$.

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 calling `GBTETH`. For more information on the sampling procedure see `[PHYS341]`.

Origin : L.Urbán
Revision :
Documentation :

Submitted: 24.03.86
Revised: 16.12.93

Total cross-section and energy loss for e^-e^+-pair production by muons

1 Subroutines

CALL **GPRELA**

`GPRELA` adds to the muon energy loss tables the contribution due to μ-nucleus interactions and e^+e^--pair production in which the energy of the pair is below threshold (see later). For the contribution of pair production, it calls the function `GPRELM`. The energy binning is in the array `ELOW` (common `/CGMULO/`). The following pointers are used:

`JMA = LQ(JMATE-I)`	pointer to the I^{th} material;
`JEL2 = LQ(JMA-2)`	pointer to dE/dx for $\mu^+\mu^-$.

`GPRELA` is called during initialisation by `GPHYSI`.

VALUE = **GPRELM** (Z,T,CUT)

Z	(REAL) atomic number of the material;
T	(REAL) kinetic energy of the muon;
CUT	(REAL) maximum energy for the pair considered by the routine.

`GPRELM` calculates the energy loss due to the direct production by muons of e^-e^+pairs with energy below `CUT` (`PPCUTM` in common `/GCCUTS/`). Above this cut, the direct pairs are produced as discrete processes (see [PHYS451]). `GPRELM` is called by `GPRELA`.

CALL **GPRSGA**

`GPRSGA` calculates the total cross-section for the pair production by photons and the direct pair production by muons. It tabulates the mean free path in cm as a function of medium and energy. For the direct pair production of muons, it calls the function `GPRSGM`. The energy binning is in the array `ELOW` (common `/CGMULO/`). The following pointers are used:

`JMA = LQ(JMATE-I)`	pointer to the I^{th} material;
`JPAIR = LQ(JMA-10)`	pointer to photon pair production cross-sections;
`JPAIR+NEK1`	pointer to muon pair production.

`GPRSGA` is called during initialisation by `GPHYSI`.

VALUE = **GPRSGM** (Z,T,CUT)

Z	(REAL) atomic number of the medium;
T	(REAL) kinetic energy of the muon;
CUT	(REAL) minimum energy for the e^-e^+pair.

`GPRSGM` calculates the total cross-section for muon direct pair production where the energy of the pair is greater than `CUT` (`PPCUTM` in common `/GCCUTS/`). It is called by `GPRSGA` (see [PHYS451]).

2 Method

When a muon of energy E moves in the field of an atom of charge Z, it can radiate a $\mathrm{e}^+\mathrm{e}^-$-pair (4^{th} order QED process) with differential cross-section [99]:

$$\frac{\partial^2\sigma}{\partial\nu\,\partial\rho} = \alpha^4\frac{2}{3\pi}(Z\lambdabar)^2\frac{1-v}{v}\left[\phi_e + \left(\frac{m_e}{m_\mu}\right)^2\phi_\mu\right] \tag{1}$$

where

$$\nu = \frac{E^+ + E^-}{E} \quad \text{and} \quad \rho = \frac{E^+ - E^-}{E^+ + E^-}$$

and

α	1/137, fine structure constant;
$\lambdabar$	3.8616×10^{-11} cm, electron Compton wavelength;
v	k/E fraction of energy transferred to the pair;
T	$E - M$ kinetic energy of the muon.
$E^\pm$	the energy of the $e^\pm$.

The explicit form of the terms ϕ_e and ϕ_μ can be found in [99]. The kinematic ranges of ν and ρ are:

$$\begin{aligned} \frac{4m_e}{E} = \nu_{min} \quad \leq \quad \nu \quad \leq \quad \nu_{max} \quad &= \quad 1 - 0.75\sqrt{e}\frac{m_\mu}{E}Z^{1/3} \\ 0 = \rho_{min} \quad \leq \quad |\rho(\nu)| \quad \leq \quad \rho_{max}(\nu) \quad &= \quad \left[1 - \frac{6m_\mu^2}{E^2(1-\nu)}\right]\sqrt{1-\frac{4m_e}{\nu E}} \end{aligned} \tag{2}$$

where $e = 2.718\ldots$.

E_c (PPCUTM in the program) is the energy cut-off; below this energy $\mathrm{e}^+\mathrm{e}^-$-pair are treated as continuous energy loss, and above they are explicitly generated and $v_c = E_c/E$. The mean value of the energy lost by the incident muon due to $\mathrm{e}^+\mathrm{e}^-$-pair with energy below E_c is:

$$E_{loss}^{pair}(Z,T,E_c) = 2E\int_{\nu_{min}}^{\nu_c} d\nu\ \nu\int_0^{\rho_{max}(\nu)} d\rho\ \frac{\partial^2\sigma}{\partial\nu\,\partial\rho} \qquad \text{GeV barn/atom} \tag{3}$$

whereas the total cross-section for the emission of a hard $\mathrm{e}^+\mathrm{e}^-$-pair is:

$$\sigma(Z,T,E_c) = 2\int_{\nu_c}^{\nu_{max}} d\nu\ \int_0^{\rho_{max}(\nu)} d\rho\ \frac{\partial^2\sigma}{\partial\nu\,\partial\rho} \qquad \text{barn/atom} \tag{4}$$

2.1 Parameterisation of energy loss and total cross-section

Instead of using the explicit formula (3) and (4), we have chosen to parameterise directly $E_{loss}^{pair}(Z,T,E_c)$ and $\sigma(Z,T,E_c)$ as:

$$\begin{aligned} \sigma(Z,T,E_c) &= Z[Z + \xi_\sigma(1+\gamma\ln Z)]\left[\ln\frac{E_{max}}{E}\right]^\alpha F_\sigma(Z,X,Y) \qquad (5) \\ E_{loss}^{pair}(Z,T,E_c) &= Z[Z + \xi_l(1+\delta\ln Z)]E\left[\frac{E_c - E_c^{min}}{E}\right]^\beta F_l(Z,X,Y) \qquad (6) \end{aligned}$$

where $\xi_{\sigma,l}$, α, β, γ and δ are parameters obtained by a fitting procedure and $E_c^{min} = E\,v_{min} = 4m_e$ and $E_{max} = E\,v_{max}$

The functions F_i (Z,X,Y) ($i = \sigma,\ l$) have the form:

$$F_i(Z,X,Y) = F_{i0}(X,Y) + ZF_{i1}(X,Y) \tag{7}$$

where $F_{ij}(X,Y)$ denotes a function constructed from two polynomials

$$F_{ij}(X,Y) = \begin{cases} P_{ij}^{neg}(X,Y) & \text{if} \quad Y \leq 0 \\ P_{ij}^{pos}(X,Y) & \text{if} \quad Y > 0 \end{cases} \tag{8}$$

and P_{ij}^{neg}, P_{ij}^{pos} fulfil the conditions:

$$P_{ij}^{neg}(X,Y)_{Y=0} = P_{ij}^{pos}(X,Y)_{Y=0} \qquad\qquad \left.\frac{\partial P_{ij}^{neg}}{\partial Y}\right|_{Y=0} = \left.\frac{\partial P_{ij}^{pos}}{\partial Y}\right|_{Y=0}$$

The detailed form of the P_{ij} polynomials is:

$$\begin{aligned} P_{i0}^{neg}(X,Y) &= (C_1 + C_2X + \cdots + C_6X^5) + (C_7 + C_8X + \cdots + C_{12}X^5)Y \\ &+ (C_{13} + C_{14}X + \cdots + C_{18}X^5)Y^2 + \cdots + (C_{31} + \cdots + C_{36}X^5)Y^5 \\ P_{i0}^{pos}(X,Y) &= (C_1 + C_2X + \cdots + C_6X^5) + (C_7 + C_8X + \cdots + C_{12}X^5)Y \\ &+ (C_{37} + C_{38}X + \cdots + C_{42}X^5)Y^2 + \cdots + (C_{55} + \cdots + C_{60}X^5)Y^5 \\ P_{i1}^{neg}(X,Y) &= (C_{61} + C_{62}X + \cdots + C_{65}X^4) + (C_{66} + C_{67}X + \cdots + C_{70}X^4)Y \\ &+ (C_{71} + C_{72}X + \cdots + C_{75}X^4)Y^2 + \cdots + (C_{81} + \cdots + C_{85}X^4)Y^4 \\ P_{i1}^{pos}(X,Y) &= (C_{61} + C_{62}X + \cdots + C_{65}X^4) + (C_{66} + C_{67}X + \cdots + C_{70}X^4)Y \\ &+ (C_{86} + C_{87}X + \cdots + C_{90}X^4)Y^2 + \cdots + (C_{96} + \cdots + C_{100}X^4)Y^4 \end{aligned} \tag{9}$$

with

$$X = \ln\frac{E}{m_\mu} \qquad\qquad Y = \ln\frac{E_c}{v_jE} \qquad\qquad j = \sigma, l$$

where $v_{\sigma,l}$ are additional parameters of the fit. By a numerical (twofold) integration of the formulae (5) and (6) above, we have calculated 1800 *data points* in the range $Z = 1, 6, 13, 26, 50, 82, 92$ and the energy ranges $1\text{GeV} \leq T \leq 10$ TeV and $4m \leq E_c \leq T$ and performed a least-squares fit to determine the parameters. The fitted values of the parameters are in the `DATA` statements in the functions `GPRSGM` and `GPRELM`, which compute the formulae (5) and (6) respectively. The accuracy of the fit is:

$$\frac{\Delta\sigma}{\sigma} \leq \begin{cases} 10\% & \text{if} \quad T \leq 5 \quad \text{GeV} \\ 5\% & \text{if} \quad T > 5 \quad \text{GeV} \end{cases}$$

$$\frac{\Delta E_{loss}^{pair}}{E_{loss}^{pair}} \leq \begin{cases} 10\% & \text{if} \quad T \leq 5 \quad \text{GeV} \\ 4\% & \text{if} \quad T > 5 \quad \text{GeV} \end{cases}$$

The function `GPRELM` contains a second formula to calculate the total energy lost by the muon due to direct e^+e^--production used when $E_c \geq E_{\max} = E - 0.75\sqrt{e}m_\mu Z^{1/3}$. This formula describes the total energy loss with an error less than 1%:

$$\begin{aligned} E_{loss}^{pair}(Z,T) &= E_{loss}^{pair}(Z,T,E_c = E_{max}) \\ &= Z(Z+1)E_{max}[d_1 + (d_2X + d_3Y) + (d_4X^2 + d_5XY + d_6Y^2) + \\ &+ \cdots + (d_{22}X^6 + d_{23}X^5Y + \cdots + d_{28}Y^6)] \end{aligned} \tag{10}$$

where $X = \ln(E/m_\mu)$ and $Y = Z^{1/3}$. The fitted parameters d_i can be found in the `DATA` statement in the function `GPRELM`.

Origin : L.Urbán
Revision :
Documentation :

Submitted: 26.10.84
Revised: 16.12.93

Simulation of e^+e^--pair production by muons

1 Subroutines

CALL **GPAIRM**

`GPAIRM` generates the e^+e^--pair radiated by a high energetic muon. It uses the following input and output:

`input:` via common block `/GCTRAK/`
`output:` via common block `/GCKING/`

`GPAIRM` is called automatically from the tracking routine `GTMUON` if, and when, the parent muon reaches its radiation point during the tracking.

2 Method

The double differential cross-section for the process can be written [99]:

$$\frac{d^2\sigma}{d\nu d\rho} = \alpha^4 \frac{2}{3\pi}(Z\lambda)^2 \cdot \frac{1-\nu}{\nu}\left[\phi_e + (m/M)^2\phi_\mu\right] \tag{1}$$

All the quantities in the expression above are defined in `[PHYS450]`. By computing this cross-section for different (ν, ρ) points, it can be seen that:

1. the **shape** of the functions $\frac{d^2\sigma}{d\nu d\rho}$ and $\frac{d\sigma}{d\nu}\int d\rho \frac{d^2\sigma}{d\nu d\rho}$ practically does not depend on Z
2. the dominant contribution comes from **the low** ν **region:**
$$\nu_{min} = (4m/E) \leq \nu \leq 100 * \nu_{min} \tag{2}$$
3. in this low region $(d^2\sigma/d\nu d\rho)$ **is flat** as a function of ρ

Therefore, we propose the following sampling method as a rough approximation:

1. In the low ν region the differential cross-section
$$\frac{d\sigma}{d\nu} = \int d\rho \frac{d^2\sigma K}{d\nu d\sigma}$$
can be approximated as:
$$\frac{d\sigma}{d\nu} \approx \left[1 - \frac{\nu_{min}}{\nu}\right]^{1/2} \frac{1}{v^a} \qquad \text{with:} \quad a = 2 - \frac{\ln E}{10} \qquad (E \text{ in GeV}) \tag{3}$$
We can write:
$$\frac{d\sigma}{d\nu} \approx f(\nu)g(\nu) \tag{4}$$

where,

$$f(\nu) = \frac{(a-1)}{\frac{1}{\nu_c^{a-1}} - \left(\frac{1}{\nu_{max}}\right)^{a-1}} \frac{1}{\nu^a} \tag{5}$$

is the normalised distribution in the interval $[\nu_c, \nu_{max}]$ and

$$g(\nu) = \left[1 - \frac{\nu_{min}}{\nu}\right]^{1/2} \tag{6}$$

is the rejection function.

2. r_1 and r_2 being two uniformly distributed random numbers in the interval $[0, 1]$:
   - Sample ν from the distribution $f(\nu)$ as:
     $$\nu = \left(\frac{1-r_1}{\nu_c^{a-1}} \frac{r_1}{\nu_{max}^{a-1}}\right)^{\frac{1}{1-a}} \tag{7}$$
   - Accept ν if $r_2 \leq g(\nu)$ (8)
3. Then compute

$$\rho_{max}(\nu) = \left[1 - \frac{6M^2}{E^2(1-\nu)}\right] \left[1 - \frac{4m}{\nu E}\right]^{1/2} \tag{9}$$

and generate ρ uniformly in the range $[-\rho_{max}, +\rho_{max}]$.

After the successful sampling of (ν, ρ), GPAIRM generates the polar angles of the radiated e^+e^--pair with respect to an axis defined along the parent muon's momentum. Θ is assigned the approximate average value:

$$\Theta = \frac{M}{E} \tag{10}$$

ϕ^+ is generated isotropically and $\phi^- = \phi^+ + \pi$

Origin : H.C.Fesefeldt, F.Carminati
Revision :
Documentation : F. Carminati

Submitted: 20.12. 85
Revised: 16.12.93

Muon-nucleus interactions

1 Subroutines

`CALL` **GMUNUI**

`GMUNUI` computes and stores in the appropriate bank the value of the muon-nucleus cross-section for a given material. It is called at initialisation time by `GPHYSI`.

`CALL` **GMUNU**

`GMUNU` 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 record) is equal to 1. 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 of the value of `IHADR`.

Note: `IMUNU` should be set to 1 only when using the `GHEISHA` hadronic interactions package. Setting it to 1 when using `FLUKA` can give unpredictable results.

`VALUE =` **GMUSIG** `(E,E1,COSTET)`

`E` (`REAL`) initial μ energy;
`E1` (`REAL`) final μ energy;
`COSTET` (`REAL`) μ scattering angle.

This function returns the value of the differential cross-section in millibarns for a muon of energy `E` to generate a nuclear interaction recoil with energy `E1` at an angle the cosine of which is `COSTET`.

2 Method

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` style of the corresponding code from the `GHEISHA` Monte Carlo Program. The `GHEISHA` routines `CASMU` and `CALIM` have been used [1].

The information contained in this chapter is mainly taken from the `GHEISHA` manual to which the user is referred. The muon-nucleus inelastic cross-section is taken as 0.0003 mb for a nucleon when the energy of the incoming muon is $E < 30$ GeV, and slowly increases 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:

$$\begin{aligned} \Gamma &= \frac{k\alpha}{2\pi}\frac{W}{E}\frac{1}{1-\epsilon} \\ \epsilon &= \left[1 + 2\frac{Q^2+\nu^2}{Q^2}\tan^2\theta^2\right] \end{aligned}$$

Q^2 and ν are the normal scaling variables expressed by:

$$Q^2 = -q^2 = 2(EW - |p||p'|\cos\theta - m^2_{-\mu}) \quad \text{and} \quad \nu = E - W$$

here σ_T and σ_S are the photo-absorption cross-sections for transverse and longitudinal photons respectively for which the relation used is:

$$\sigma_S = 0.3\left(1 - \frac{1}{1.868}Q^2\nu\right)\sigma_T$$

and σ_T is assumed to be constant $\sigma_T = 0.12$ mb. For the incident flux K of the photons Gillman's convention is used:

$$K = \nu + Q^2/2\nu$$

A three-dimensional importance sampling in the variables E, W and θ is performed each time an interaction has to occur.

This set of routines only works if GUPHAD calls GPGHEI and GUHADR calls GHEISH. 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 π-nucleus inelastic scattering. While the final state generated this way gives a good approximation for calorimetric purposes, the kinematics of the final state may be a rather poor approximation of reality. The muon-nucleus interactions are activated by the MUNU data record 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 initialisation time. See material bank structure for details.

Origin : H.C. Fesefeldt, F.Carminati
Revision :
Documentation : F. Carminati

Submitted: 24.02.86
Revised: 11.10.93

The GEANT/GHEISHA Interface

1 Introduction

There are two packages to describe the hadronic showers in matter which are available to the users of GEANT: GHEISHA [1] and FLUKA (see [PHYS520]).

The GHEISHA code generates hadronic interactions with the nuclei of the current tracking medium, evaluating cross-sections and sampling the final state multiplicity and kinematics, while the GEANT philosophy is preserved for the tracking. The GHEISHA code is stored in the GEANH car file.

The GHEISHA printing flags can be set via the GEANT SWIT data record: each switch greater than 100 but smaller that 111 sets the corresponding printing flag of GHEISHA modulo 100, so that SWIT 105 will set the printing flag 5 of GHEISHA. Setting the printing flags of GHEISHA the following information will be displayed:

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 suspected errors. NPRT(8) produces the most illustrative output. Those flags work in conjunction with the DEBUG data record of GEANT.

1.1 Description of the routines

CALL **GHESIG** (P,EK,AVER,A,Z,W,NLM,DENS,CORR,IPART)

P (REAL) momentum of the particle (in GeV/c);
EK (REAL) kinetic energy of the particle (in GeV);
AVER (REAL) average mass number of the material;
A(NLM) (REAL) vector containing the mass numbers of the components of the mixture, the same as AVER in case of an element;
Z(NLM) (REAL) vector containing the atomic numbers of the components of the mixture, or the atomic number in case of an element;
W(NLM) (REAL) vector of length NLM containing the relative weights of the components of the mixture (normalised to one), one in case of an element;
NLM (INTEGER) number of components of the mixture, 1 in case of an element;
DENS (REAL) density of the material;
CORR (REAL) if this parameters is > 0, then corrections are applied to the cross-section; should be used in case of inorganic scintillators;
IPART (INTEGER) GEANT particle code.

This routine returns the total macroscopic cross-section in cm^{-1}. The correction flag is taken from the 26^{th} word of the *next* bank in the tracking medium linear structure pointed at by `LQ(JTMED-NUMED)`.

The cross-sections on nucleus are known only for pions and protons. The general law:

$$\sigma(A) = 1.25 * \sigma_{tot}^{Proton}\ A^{\alpha}$$

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 H, Al, Cu and Pb the measured cross-sections are stored in `DATA` statements.

The stored cross-sections are the pre-calculated `GHEISHA` ones. As a starting point the measured cross-sections of pions, kaons, protons, antiprotons and neutrons on protons are used. The cross-sections tabulated are measured values taken from the CERN HERA compilations. The values for $\mathrm{K}_s^0/\mathrm{K}_l^0$ are updated as of 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. All the cross-sections are contained in data statements so no external file is needed for `GHEISHA`.

`CALL` **GPGHEI**

This routine returns the distance to the next hadronic interaction according to the `GHEISHA` cross-sections. It calls `GHESIG` and is called by `GUPHAD`. The default copy provided in the `GEANT` library uses the `GHEISHA` shower code:

```
       SUBROUTINE GUPHAD
*
*****************************************************************
*                                                               *
*   GEANT  user routine called at each step to evaluate         *
*   the remaining distance to the hadronic interaction point    *
*                                                               *
*****************************************************************
*
      CALL GPGHEI
*
      END

      SUBROUTINE GUHADR
*.
****************************************************************
*                                                              *
*       GEANT  user routine called when a hadronic process     *
*       has been selected in the current step, in order to     *
*       generate the final particle's state                    *
*                                                              *
****************************************************************
*
       CALL GHEISH
*
       END
```

`CALL` **GHEISH**

This is the main steering routine for the hadronic interactions and is a fan-out to the various *cascade* routines of `GHEISHA` which treat the particular hadronic interaction. Here the kind of interaction is stored in the `INT` flag, with the following meaning:

`INT=0`	no interaction (`NONE`);
`INT=1`	elastic scattering occurs (`ECOH`);
`INT=2`	inelastic incoherent interaction occurs (`INHE`), 1 and 2 include also nuclear reaction processes at very low energies;
`INT=3`	nuclear fission with inelastic scattering occurs (`FISS`);
`INT=4`	neutron nuclear capture occurs (`CAPT`).

After the interaction has been selected, the appropriate cascade routine is called. Upon exit from this there is a check whether the interaction has generated new particles or not. If yes, the new particles are copied in the `GEANT` temporary stack (`GKING`). If the particle is a heavy fragment or a proton and it is below the energy cut specified via the `CUTS` data record, it is not stored in the stack but the kinetic energy is collected. The size of the `GHEISHA` stack is parametrised, and its limit is currently set to 100 in sequence `GCKMAX`. The user is left to decide in `GUSTEP` what to do with the new tracks. This routine is called also in case of a stopping hadron, i.e. with kinetic energy below the `GEANT` cuts. In this case the routine `GHSTOP` (see later) is called to handle the stopping hadron. The printing flags for `GHEISHA` are also set in this routine according to the current value of `IDEBUG`.

This routine must be called by the user routine `GUHADR` (see above).

As explained above, for inorganic scintillators such as the BGO, it is possible to activate a correction to the hadronic cross-section. This is done via the routine `GSTPAR` in the following way:

```
CALL GSTPAR(ITMED,'GHCOR1',1.0)
```

The 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 used in `GSTMED` for which corrections are requested. This routine has to be called before `GPHYSI`.

`CALL` **GHSTOP**

This is an internal routine used to handle stopping particles, called by `GHEISH`. Here again we have a switch to the various routines handling stopping particles. In particular this routine can lead to nuclear absorption for negative pions and negative kaons (`ABSO`) and to annihilation for antineutrons, and antiprotons (`ANNH`). The kinetic energy is completely absorbed, and in the case of an unstable particle, the particle is decayed at rest via the standard `GEANT` decay routine `GDECAY`.

Origin : A.Ferrari, K.Lassila-Perini
Revision :
Documentation : K.Lassila-Perini

Submitted: 25.10.91
Revised: 17.12.93

The GEANT/FLUKA Interface

1 Subroutines

CALL **FLINIT**

FLINIT initialises the FLUKA variables and reads data from a file (flukaaf.dat) which is automatically opened. FLINIT is called from FLDIST when a hadron enters there first time in the run.

CALL **FLDIST**

FLDIST computes the distance to the next interaction point. It calls the FLUKA routines to compute the cross-sections for all particles except neutrons with kinetic energy below 20 MeV for which GHEISHA routines are called. FLDIST is called from the user routine GUPHAD where the hadronic package can be chosen.

CALL **FLUFIN**

FLUFIN calls the FLUKA routines to generate the hadronic interaction. It passes the particle to FLUKA interaction routines and puts the eventual secondary particles to the GEANT stack. FLUFIN is called from the user routine GUHADR.

2 Method

FLUKA [2, 3, 4, 5, 6, 101, 102] is a simulation program which as a standalone code contains transport and the physical processes for hadrons and leptons and tools for geometrical description. In GEANT, only the hadronic interaction part is included.

The total cross-section of the hadronic processes is computed by FLUKA routines called from FLDIST (the cross-section for neutrons below 20 MeV is computed in GHEISHA). If hadronic intercation is chosen in GEANT tracking routine, the particle is passed to FLUFIN.

If particles are stopping (i.e. their energy is below the cut-off energy), their kinetic energy is deposited. However, if the particle can decay (π^+, $K^\pm$) it is forced to decay, or if it is an annihilating particle (π^-, $\bar{n}$, $\bar{p}$), it is sent to FLUKA routines for annihilation. The neutrons with kinetic energy below 20 MeV are passed to GHEISHA.

If the particle is not stopping, a sampling is done between elastic and inelastic processes. The cross-sections have been computed in FLDIST in the same time as the total cross-section. The particle is sent correspondingly to the elastic or inelastic interaction routines. After the interaction, the eventual secondary particles are written to GEANT stack. The program flow is shown in figures 46 and 45.

When the tracking media is a mixture or a compound material (defined by GSMIXT, see [CONS110]), the atom with which the interaction is taking place is chosen by sampling on the basis of the cross-sections. This is important especially in hydrogenous materials.

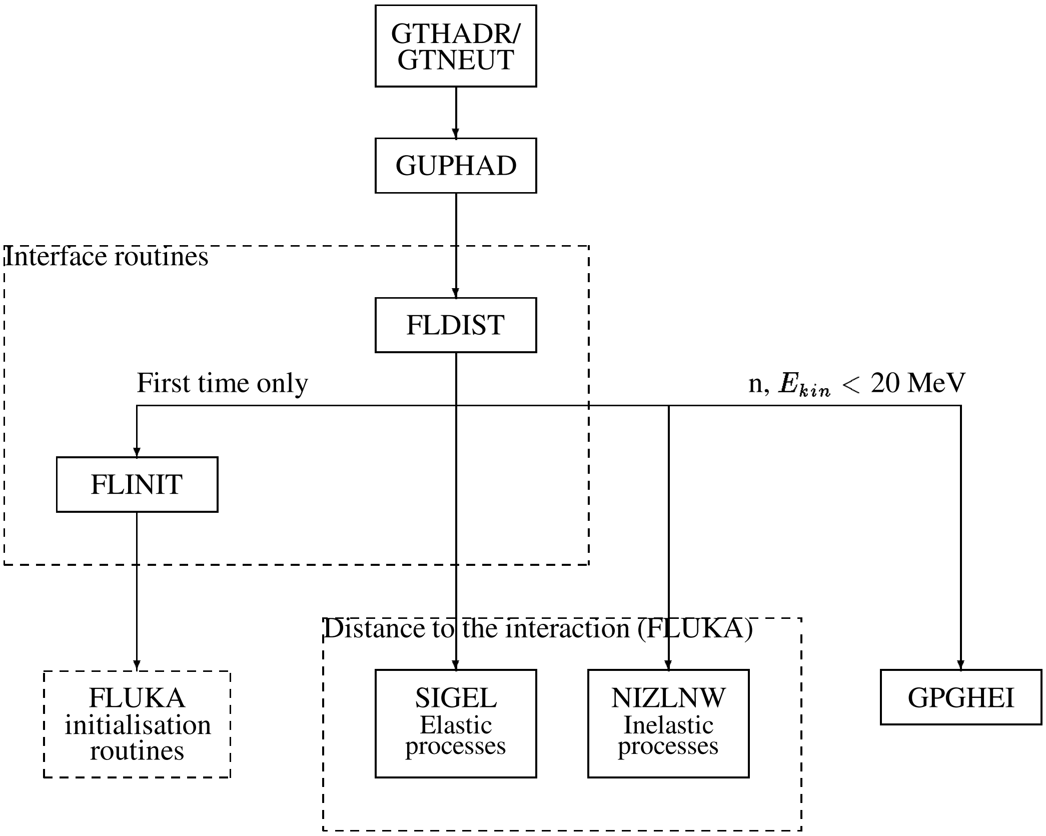


Figure 45: Program flow for calculation of the distance to the next interaction point

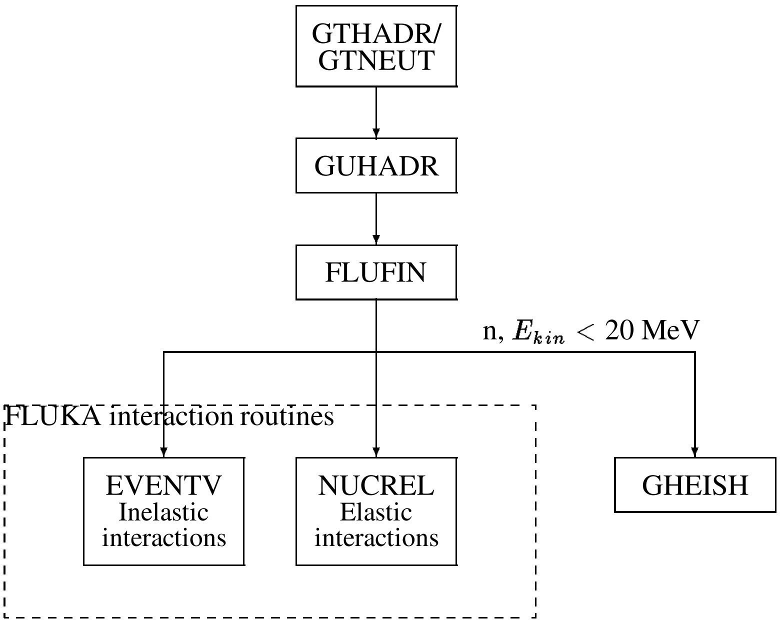


Figure 46: Program flow for generating secondary particles

Origin : T.Gabriel, C.Zeitnitz, K.Lassila-Perini
Revision :
Documentation : K.Lassila-Perini

Submitted: 17.12.93
Revised: 17.12.93

The GEANT/MICAP interface

1 Subroutines

CALL **GMORIN**

GMORIN initialises the MICAP variables and reads the cross-section file for the materials that have been defined. It is called from GFMDIS when a neutron with kinetic energy below 20 MeV enters there first time.

CALL **GFMDIS**

GFMDIS computes the distance to the next interaction points. It calls the FLUKA routines to compute the cross-sections for all particles except neutrons with kinetic energy below 20 MeV for which MICAP function SIGMOR is called. GFMDIS is called from the user routine GUPHAD where the hadronic package can be chosen. The only difference between GFLDIS (see [PHYS520]) and GFMDIS is that for the former, GHEISHA hadronic package is called for the neutrons below 20 MeV, and for the latter, the low-energy neutrons are handled by MICAP.

CALL **GFMFIN**

GFMFIN calls the FLUKA routines to generate the hadronic interaction. For the neutrons with kinetic energy below 20 MeV GMICAP is called. The only difference between GFLFIN (see [PHYS520]) and GFMFIN is that for the former, GHEISHA hadronic package is called for the neutrons below 20 MeV, and for the latter, the low-energy neutrons are handled by MICAP. GFMFIN is called from the user routine GUHADR.

CALL **GMICAP**

GMICAP calls the MICAP routines to handle the low-energy interaction of neutrons. It writes the eventual secondaries to the GEANT stack. GMICAP is called from GFMFIN.

2 Method

MICAP (A Monte Carlo Ionization Chamber Analysis Package) [103], [104] is a Monte Carlo system to analyze ionisation chamber responses. As a standalone program it contains the code for formatting the cross-section files, neutron and photon transport, the geometry definitions and the code for the chamber response. In GEANT, only the sampling of the neutron interactions from the already prepared cross-section file is included. The interface between GEANT and MICAP has been extracted from GCALOR package [105] by C.Zeitnitz and T. Gabriel.

When using GEANT-MICAP interface the low-energy neutrons are handled in MICAP routines. Other hadrons and high-energy neutrons are passed to FLUKA interaction routines. For low-energy neutrons, the total cross-section is given by MICAP and if the neutron interaction is chosen by GEANT tracking routine, GMICAP reads the cross-section for neutron interactions processes, samples and generates the interaction and the returns the secondary particles (nucleons, heavy fragments, or photons) to GEANT. Information on the recoil nucleus (atomic number AMED, charge ZMED and kinetic energy ERMED) can be found in MCRECO common block. The program flow is shown in figures 48 and 47.

MICAP uses pointwise cross-section data (as a difference to so called group cross-sections where the data are averaged over certain energy intervals). This method has the advantage that the resonances are not smoothed by averaging the data. The neutron cross-section are available for the following isotopes:

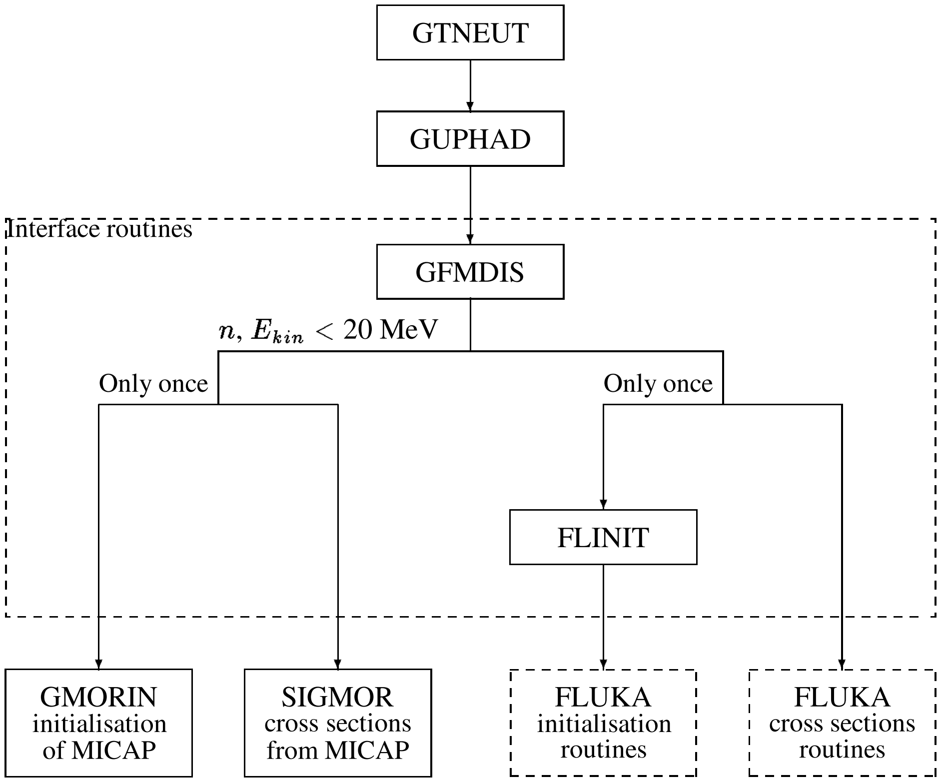


Figure 47: Program flow for calculation of the distance to the next interaction point

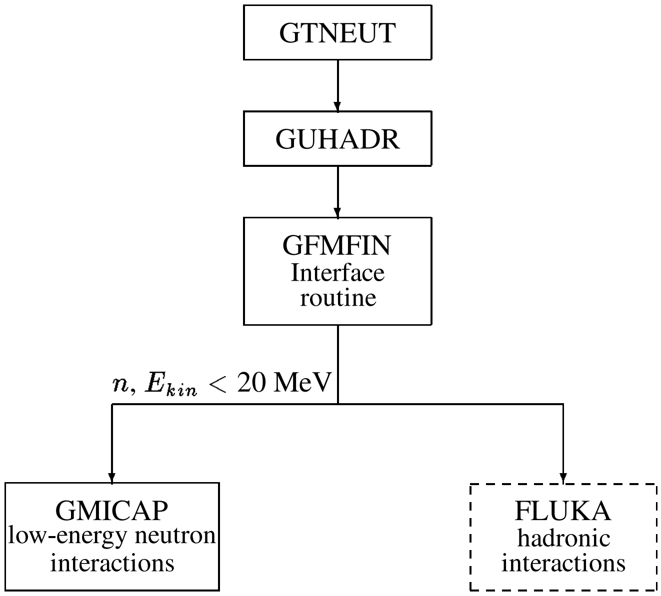


Figure 48: Program flow for generating secondary particles

Hydrogen (bound)	Argon
Hydrogen (free)	Calcium
Lithium (6,7)	Chromium
Boron (10,11)	Iron
Carbon	Nickel
Nitrogen	Copper
Oxygen	Molybdenum
Fluorine	Copper
Sodium	Molybdenum
Magnesium	Barium
Aluminum	Tantalum
Silicon	Lead
Chlorine	Uranium (235,238)

If the cross-sections are not found for some of the defined materials, a warning is printed first at the initialisation time telling which cross-section are used (the closest Z available) instead. Then, an additional warning is printed each tracking step.

PHYS Bibliography

[1] H.C.Fesefeldt. Simulation of hadronic showers, physics and applications. Technical Report PITHA 85-02, III Physikalisches Institut, RWTH Aachen Physikzentrum, 5100 Aachen, Germany, September 1985.

[2] P.A.Aarnio et al. Fluka user's guide. Technical Report TIS-RP-190, CERN, 1987, 1990.

[3] A.Fassò A.Ferrari J.Ranft P.R.Sala G.R.Stevenson and J.M.Zazula. FLUKA92. In *Proceedings of the Workshop on Simulating Accelerator Radiation Environments*, Santa Fe, USA, 11-15 January 1993.

[4] A.Fassò A.Ferrari J.Ranft P.R.Sala G.R.Stevenson and J.M.Zazula. A Comparison of FLUKA Simulations with measurements of Fluence and Dose in Calorimeter Structures. *Nuclear Instruments & Methods A*, 332:459, 1993.

[5] C.Birattari E.DePonti A.Esposito A.Ferrari M.Pelliccioni and M.Silari. Measurements and characterization of high energy neutron fields. *approved for pubblication in Nuclear Instruments & Methods A*.

[6] P.A.Aarnio, A.Fassò, A.Ferrari, J.-H.Moehring, J.Ranft, P.R.Sala, G.R.Stevenson and J.M.Zazula. FLUKA: hadronic benchmarks and applications. In *MC93 Int. Conf. on Monte-Carlo Simulation in High-Energy and Nuclear Physics*, Tallahassee, Florida, 22-26 February 1993. Proceedings in press.

[7] Particle Data Group. Review of particle properties. *Physical Review D*, 45(11), June 1992.

[8] R.Brun, M.Goossens, and J.Zoll. *ZEBRA Users Guide,* Program Library Q100. CERN, 1991.

[9] J.H.Hubbell, H.A.Gimm, and I.Overbo. *Jou. Phys. Chem. Ref. Data*, 9:1023, 1980.

[10] J.C.Butcher and H.Messel. *Nucl. Phys*, 20:15, 1960.

[11] J.M.Hammersley and D.S.Handscomb. *Monte Carlo methods*. J.Wiley et Sons Inc, New York, 1964.

[12] R.Ford and W.Nelson. SLAC 210, UC 32. SLAC, June 1978.

[13] Y-S.Tsai. *Rev. Mod. Phys.*, 46:815, 1974.

[14] Y-S.Tsai. *Rev. Mod. Phys.*, 49:421, 1977.

[15] W.R.Nelson et al. The EGS4 code system. Technical Report 265, SLAC, 1985.

[16] O.Klein and Y.Nishina. *Z. f. Physik*, 52:853, 1929.

[17] H.Messel and D.F.Crawford. volume 3. Pergamon Press, 1970.

[18] B.Rossi. Prentice-Hall, 1952.

[19] F. Biggs and R. Lighthill. Analitycal approximations for x-ray cross sections 3. *Sandia National Lab., SAND87-0070 UC-34*, 1987.

[20] H. Bethe and E. Salpeter. *Encyclopedia of Physics*, volume XXXV. Springer-Verlag, Berlin, 1957.

[21] H. Hall. The theory of photoelectric absorbtion for x-rays and γ-rays. *Rev. Mod. Phys.*, 8:358, 1936.

[22] R. H. Pratt. Atomic photoelectric effect at high energies. *Phys. Rev.*, 117:1017, 1960.

[23] R. H. Pratt. Photoeffect form the l shell. *Phys. Rev.*, 119:1619, 1960.

[24] WM. J. Veigele. Photon cross sections from 0.1 kev to 1 mev for elements z = 1 to z = 94. *Atomic Data Tables*, 5:51, 1973.

[25] M. Gavrila. Relativistic l-shell photoeffect. *Phys. Rev.*, 124:1132, 1961.

[26] F. Sauter. *Ann. Physik*, 9:217, 1931.

[27] F. Sauter. *Ann. Physik*, 11:454, 1931.

[28] M. Gavrila. Relativistic k-shell photoeffect. *Phys. Rev.*, 113:514, 1959.

[29] B. C. H. Nagel and P. O. M. Olsson. *Arkiv f. Fysik*, 18:1, 1960.

[30] M. O. Krause. Atomic radiative and radiationless yields for k and l shells. *J. Phys. Chem. Ref. Data*, 8:307, 1979.

[31] S.T. Perkins et al. *UCRL-50400*, 30, 1991.

[32] J. H. Scofield. Relativistic hatree-slater values for k and l x-ray emission rates. *Atomic Data and Nuclear Data Tables*, 14:121, 1974.

[33] J.H.Hubbell and I.Overbo. *J. Phys. Chem. Ref. Data*, 1/8, 1979.

[34] H.Storm and H.I.Israel. Nucl. data tables. A7:565, 1970.

[35] W.R.Nelson, H.Hirayama, and D.W.O.Rogers. Technical Report 265, SLAC, 1985.

[36] J.D.Jackson. *Classical Electrodynamics*. J.Wiley et Sons, Inc, New York, 1975.

[37] H.A.Bethe. *Phys. Rev.*, 89:1256, 1953.

[38] B.Rossi. Prentice-Hall, Englewood Cliffs, 1961.

[39] B.Rossi and K.Greisen. *Rev. Mod. Physics*, 13:240, 1941.

[40] V.L.Highland. *Nucl. Inst. and Meth.*, 129:497, 1975.

[41] V.L.Highland. *Nucl. Inst. and Meth.*, 161:171, 1979.

[42] G.R.Lynch and O.I.Dahl. Approximation to multiple coulomb scattering. Technical Report LBL-28165, Physics Division, Laurence Berkeley Laboratory, University of California, Berkeley, CA 94720, February 1990.

[43] G.Z.Molière. Theorie der Streuung schneller geladener Teilchen I: Einzelstreuung am abgerschmitten Coulomb-Feld. *Z. Naturforsch.*, 2a:133, 1947.

[44] G.Z.Molière. Theorie der Streuung schneller geladener Teilchen II: Merfach- und Vielfachstreuung. *Z. Naturforsch.*, 3a:78, 1948.

[45] W.T.Scott. *Rev. Mod. Phys.*, 35:231, 1963.

[46] S.Goudsmit and J.L.Saunderson. Multiple scattering of electrons. *Phys. Rev.*, 57:24, 1940.

[47] S.Goudsmit and J.L.Saunderson. Multiple scattering of electrons II. *Phys. Rev.*, 36:36, 1940.

[48] T.M.Jenkins, W.R.Nelson, and A.Rindi. *MonteCarlo transport of electrons and photons*. Plenum Press, New York, 1988.

[49] E.Keil, E.Zeitler, and W.Zinn. Zur Einfach- und Mehrfachstreuung geladener Teilchen. *Z.Naturforsch.*, 15a:1031, 1960.

[50] S.M. Selzer M.J. Berger. Technical Report SP 3012, NASA, 1964.

[51] H.A.Bethe. *Ann. Phys.*, 5:325, 1930.

[52] F.Bloch. *Z. fur Phys.*, 81:363, 1933.

[53] R.M.Sternheimer. *Phys. Rev. B*, 3:3681, 1971.

[54] H.Messel and D.F. Crawford. Pergamon Press, Oxford, 1970.

[55] Particle Data Group. Lawrence Berkeley Laboratory and CERN, 1992.

[56] G.N.Patrick. Physical processes in the simulation of electromagetic cascades. In *Workshop on Shower Simulation for LEP Experiments*, January 1985.

[57] L. Landau. On the Energy Loss of Fast Particles by Ionisation J. Phys. 8:201, 1944. In D.ter Haar, editor, *L.D.Landau, Collected papers*, page 417. Pergamon Press, Oxford, 1965.

[58] K.S.Kölbig and B.Schorr. Asymptotic expansion for the Landau density and distribution functions. *Comp. Phys. Comm.*, 32:121, 1984.

[59] R.Talman. On the statistics of particle identification using ionization. *Nucl. Inst. Meth.*, 159:189, 1979.

[60] R.Jones. Private communication to a GEANT monthly meeting.

[61] P.V.Vavilov. Ionisation losses of high energy heavy particles. *Soviet Physics JETP*, 5:749, 1957.

[62] B.Schorr. Programs for the Landau and the Vavilov distributions and the corresponding random numbers. *Comp. Phys. Comm.*, 7:216, 1974.

[63] S.M.Seltzer and M.J.Berger. Energy loss straggling of protons and mesons. In Studies in Penetration of Charged Particles in Matter. Nuclear Science Series 39, Nat. Academy of Sciences, Washington DC, 1964.

[64] F.James. Program formerly in the CERN Program Library, but now replaced by G110, entry RANLAN.

[65] J.E.Moyal. Theory of ionisation fluctuations. *Phil. Mag.*, 46:263, 1955.

[66] K.S.Kölbig and B.Schorr. A program package for the Landau distribution. *Comp. Phys. Comm.*, 31:97, 1984.

[67] A. Rotondi and P. Montagna. Fast calculation of Vavilov distribution. *Nucl. Inst. Meth.*, B47:215, 1990.

[68] W. W. M. Allison and J. H. Cobb. *Ann. Rev. Nucl. Part. Sci.*, 30:253, 1980.

[69] S. K. Kotelnikov V. M. Grishin, V. K. Ermilova. *Nucl. Instr. Meth.*, A307:273, 1991.

[70] *At. data Nucl. data tables*, 20:1, 1978.

[71] *At. data Nucl. data tables*, 27:2, 1982.

[72] *At. data Nucl. data tables*, 32:1, 1985.

[73] J.B.Birks. *The theory and practice of Scintillation Counting*. Pergamon Press, 1964.

[74] R.L.Craun and D.L.Smith. *Nucl. Inst. and Meth.*, 80:239–244, 1970.

[75] S.M. Seltzer and M.J. Berger. *Nucl. Inst. Meth.*, B12:95, 1985.

[76] A.B.Migdal. *Phys. Rev.*, 103:1811, 1956.

[77] L.Kim, R.H.Pratt, S.M.Seltzer, and M.J.Berger. *Phys. Rev.*, A33:3002, 1986.

[78] R.W.Williams. *Fundamental Formulas of Physics*, volume 2. Dover Pubs. Inc, 1960.

[79] W.Heitler. *The Quantum Theory of radiation*. Clarendon Press, Oxford, 1954.

[80] D. Vaughan. *X-Ray Data Booklet*. Center for X-Ray Optics, 1986.

[81] R.Maier. In *CERN Accelerator School*. CERN, 17-28 September 1990. CERN 91-04.

[82] C.Bricman et al. *Rev. Mod. Phys.*, 52, 1980.

[83] H.H.Andersen and J.F.Ziegler. *Hidrogen stopping powers and ranges in all elements*. Pergamon Press, 1977.

[84] R.M.Sternheimer. *Phys. Rev.*, 88:851, 1952.

[85] R.M. Sternheimer. *Phys. Rev.*, 183:511, 1956.

[86] W.H.Barkas. Technical Report 10292, UCRL, August 1962.

[87] C.R.Serre. Technical Report 67 5, CERN, March 1967.

[88] C.R.Serre. Technical Report 71 18, CERN, September 1971.

[89] H.K.Reynold et al. *Phys. Rev.*, 92:742, 1953.

[90] D.W.Green et al. *Phys. Rev.*, 98:466, 1955.

[91] N.Sakamoto et al. *Rad. Effects and Defects in Solids*, 117:193, 1991.

[92] W.H.Barkas and M.J. Berger. Technical Report SP 3013, NASA, 1964.

[93] H.H.Andersen and J.F.Ziegler. *Helium stopping powers and ranges in all elements*. Pergamon Press, 1977.

[94] W.Heinrich et al. Technical Report GSI-91-30, GSI, Darmstad, June 1991.

[95] F.Hubert et al. *At. Data Nucle. Data Tables*, 46:1, 1990.

[96] C.Tschalär and H.D.Maccabee. *Phys. Rev. B*, 1:2863, 1970.

[97] H.Geisel et al. *NIM*, 194:21, 1982.

[98] H.Schmidt-Böring and H.Harnung. *Z.f.Phys.*, A286:253, 1978.

[99] W. Lohmann and R. Voss. Yellow Report 85 03, CERN, 1985.

[100] P.Marmier and E.Sheldon. volume 1. Academic Press, 1969.

[101] P.A.Aarnio, A.Fassò, A.Ferrari, J.-H.Moehring, J.Ranft, P.R.Sala, G.R.Stevenson and J.M.Zazula. Electron-photon transport: always so good as we think? Experience with FLUKA. In *MC93 Int. Conf. on Monte-Carlo Simulation in High-Energy and Nuclear Physics*, Tallahassee, Florida, 22-26 February 1993. Proceedings in press.

[102] A.Ferrari and P.R.Sala. A New Model for hadronic interactions at intermediate energies for the FLUKA code. In *MC93 Int. Conf. on Monte-Carlo Simulation in High-Energy and Nuclear Physics*, Tallahassee, Florida, 22-26 February 1993. Proceedings in press.

[103] J.O. Johnson and T.A. Gabriel. A User's Guide to MICAP: A Monte Carlo Ionization Chamber Analysis Package. Technical Report TM-10340, ORNL, 1988.

[104] J.O. Johnson and T.A. Gabriel. Development and Evaluation of a Monte Carlo System for Analysis of Ionization Chamber Responses. Technical Report TM-10196, ORNL, 1987.

[105] C.Zeitnitz T.Gabriel. In *Proc. of International Conference on Calorimetry in High Energy Physics*, Tallahasse, FL, USA, February 1993.

Origin :
Revision :
Documentation : F.Bruyant

Submitted: 15.08.84
Revised: 16.12.93

The tracking package

1 Introduction

In GEANT *tracking* a particle through a geometry of objects consists in calculating a set of points in a seven-dimensional space (x, y, z, t, p_x, p_y, p_z) which is called the trajectory of the particle. This is achieved by integrating the equations of motion over successive *steps* from one trajectory point to the next and applying corrections to account for the presence of matter.

To have a detailed description of the kinematic of the particle it would be necessary to calculate a trajectory point every time the component of the momentum change. This is not possible because it would mean calculating an enormously large number of points. Processes like the deviation of a charged particle in a magnetic field, the loss of energy due to bremsstrahlung and ionisation or the deviation due to elastic electromagnetic scatterings are essentially continuous. An arbitrary distinction is thus made between discrete and continuous processes, controlled by a set of thresholds which can be set by the user. A particle trajectory is thus a set of points at which a discrete process has happened.

The tracking package contains a subprogram which performs the tracking for all particles in the current event and for the secondary products which they 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 structure one of the critical tasks is the estimation *a priori* of the step size. This is performed automatically by the program. For a particle with a 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 the (quasi)continuous processes which usually impose a limit to the interval of integration (energy loss, multiple scattering ...) or to the occurrence of a discrete process which introduces a discontinuity in the trajectory (decay, electromagnetic or hadronic interaction). The modification of the kinematic parameters (position and energy) due to continuous processes between two discrete ones is taken into account and the necessary modifications are applied at the end of the step.

In addition to the 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 the program will set automatically, but which may be also optimised by the user, such as:

- the maximum turning angle due to magnetic field permitted in one step;
- the maximum fractional energy loss in one step;
- the maximum geometrical step allowed;
- the accuracy for crossing medium boundaries;
- the minimum step size due to either energy loss or multiple scattering.

These quantities are part of the so-called tracking medium parameters. They can either be calculated by the program or provided by the user and are stored in the data structure JTMED, through the routine GSTMED (see

[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 information required for the computation of the step size is found 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, GTMEDI, GTNEXT and GINVOL (see [GEOM]). Some information is computed at tracking time such as the probability of occurrence of an interaction.

For convenience every particle is assigned a tracking type:

1. γ (GTGAMA);
2. $e^{\pm}$ (GTELEC);
3. neutral hadrons and neutrinos (GTNEUT);
4. charged hadrons (GTHADR);
5. muons (GTMUON);
6. geantinos, particles only sensitive to geometry used for debugging (GTNINO);
7. Čerenkov photons (GTCKOV);
8. heavy ions (GTHION).

Which processes have 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 GHEISHA, FLUKA have been selected (see [PHYS001]).

3 Transport subroutines

At the event level the tracking is controlled by the subroutine GTREVE called by the subroutine GUTREV where the user can perform additional actions. GTREVE loops over all vertices and for every vertex in turn stores all tracks in the stack JSTAK. Then tracking begins: particles are fetched from JSTAK by GLTRAC which prepares the commons for tracking. GUTRAK is then called, which calls GTRACK. The subroutine GTRACK transports the particle up to the its 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 the JSTAK stack, and possibly in the permanent structure JKINE.

The subroutine GTRACK transports the track through the geometrical volumes identifying, through the subroutine GTMEDI, every new volume which the particle has reached, and storing the corresponding material and tracking medium constants in the common blocks /GCMATE/ and /GCTMED/. The actual transport is performed by a different routine for each tracking type, as indicated in the previous section. These compute the physical step size according to the activated physics processes, and compute the geometrical limit for the step, only when necessary, through GTNEXT, and propagate the particle over the computed step.

4 Magnetic field transport routines

Once the step size has been decided, transport proceeds along a straight lines for all neutral particles and for charged particle in absence of magnetic field. When magnetic field is present, the direction of charged particles will change along the step. The routine GUSWIM calculates the deviation in magnetic field over a given step. Depending on the tracking medium parameter IFIELD the GUSWIM routine calls:

- GRKUTA for inhomogeneous fields, IFIELD=1;
- GHELIX for quasi-homogeneous fields, IFIELD=2;

- GHELX3 for uniform fields along the z axis, IFIELD=3).

GRKUTA and GHELIX call the user subroutine GUFLD to obtain the components of the field in a given point. GHELX3 takes the value of the field from the tracking medium parameter FIELDM.

5 The subroutine GUSTEP

The current track parameters are available in the common /GCTRAK/ together with all variables necessary to 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 informs on the boundary crossing:

  0 transport inside a volume;
  1 entering a new volume or beginning of new track;
  2 exiting a volume;
  3 exiting the first mother;

- The flag ISTOP informs on the particle status:

  0 normal transport;
  1 particle has disappeared (decay, interaction...);
  2 particle has crossed a threshold (time, energy...).

- The array LMEC informs on the *mechanisms* active in the current step. The mechanism names are stored in ASCII equivalent in LMEC(1...NMEC).

- The total energy loss in 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 by GTRACK at the end of every step and when entering a new volume.

The variable NGKINE in common /GCKING/ contains the number of secondary particles generated at every step, and which are stored in the same common. Depending on the application and on the particle type the user may then take the appropriate action in GUSTEP:

- ignore the particle;
- store the secondary produced in the JSTAK stack for further tracking;
- store the secondary also in the JKINE/JVERTX structure where it will be kept till the end of the event.

6 Connection with the detector response package

The detector response package ([HITS]) allows to establish a correspondence between the volumes seen by the particle and the active 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 fills the common block /GCSETS/ with 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 debugging (subroutine GPJXYZ) or for the graphical representation of the particle trajectories [DRAW].

Origin : R.Brun
Revision :
Documentation :

Submitted: 01.10.84
Revised: 16.12.93

Steering routine to track one event

`CALL` **GTREVE**

Controls the tracking at event level.

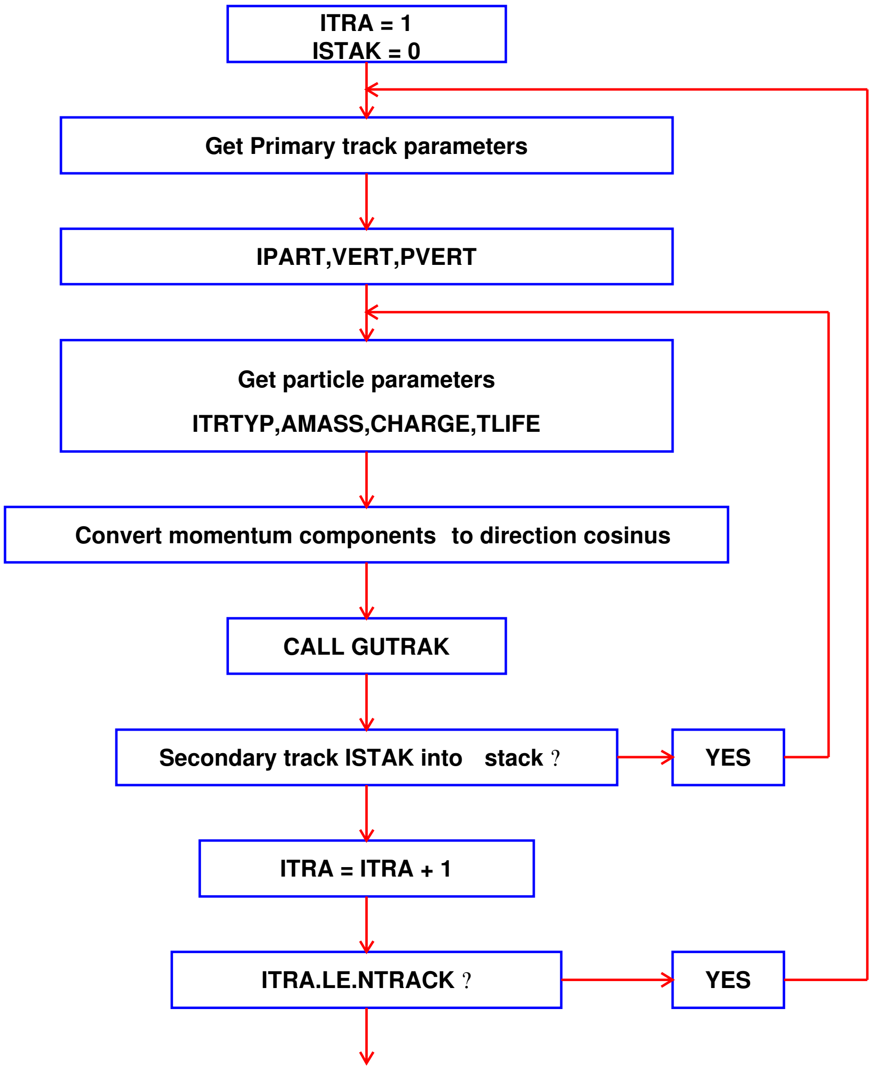


Figure 49: `GTREVE` block diagram

Origin : R.Brun,F.Bruyant,M.Maire
Revision :
Documentation :

Submitted: 01.10.84
Revised: 16.12.93

Steering routine to track one particle

`CALL` **GTRACK**

Controls the tracking at track level.

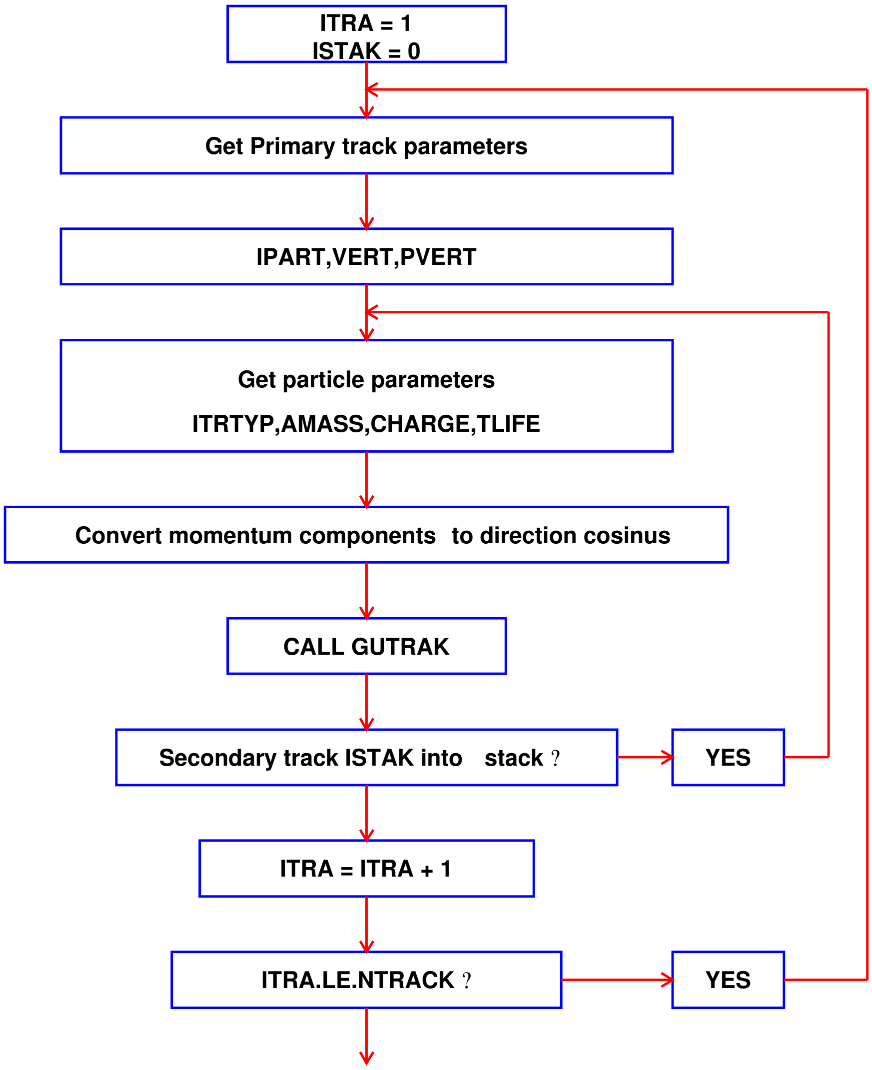


Figure 50: `GTRACK` block diagram

Origin : R.Brun,F.Bruyant,M.Maire **Submitted:** 01.10.84
Revision : **Revised:** 16.12.93
Documentation :

Tracking one particle through a volume

GTVOL block diagram.

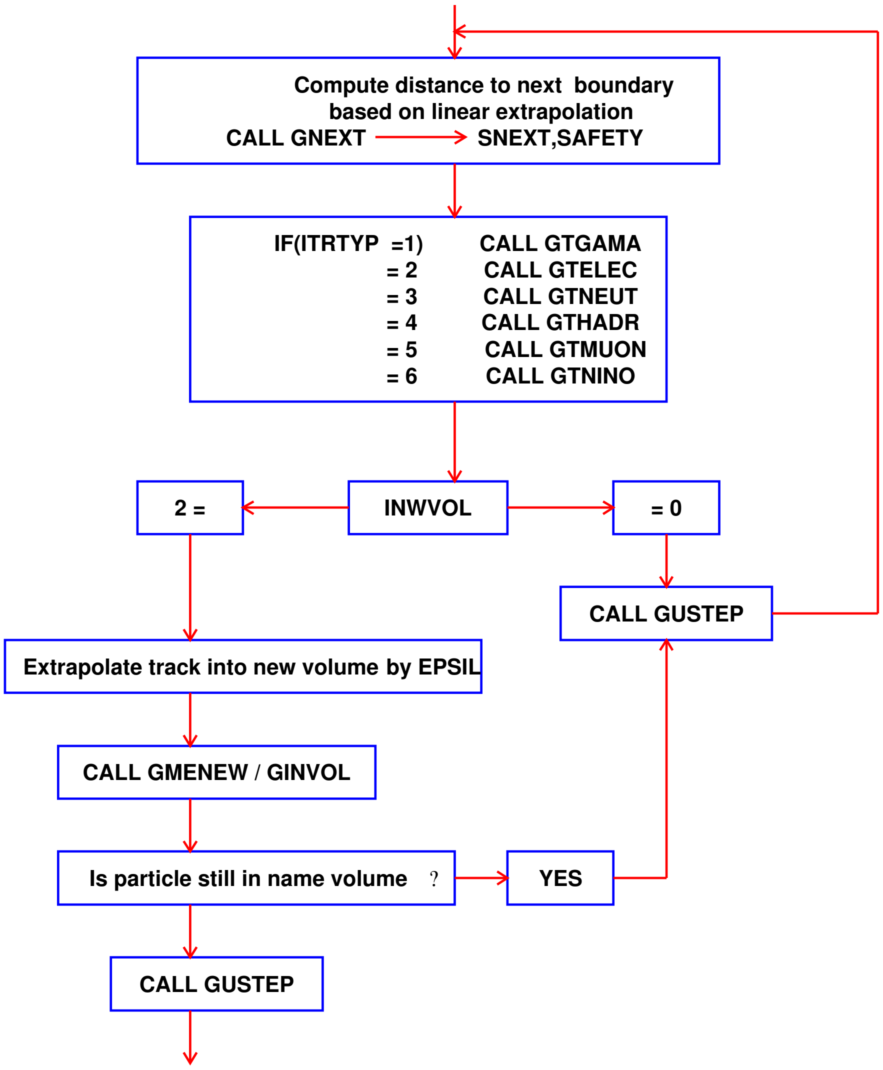


Figure 51: Example of geometrical tree structure

Origin : R.Brun, M.Maire
Revision : F.Carminati, S.Giani, L.Urbán
Documentation :

Submitted: 01.10.84
Revised: 16.12.93

The tracking routines block diagrams

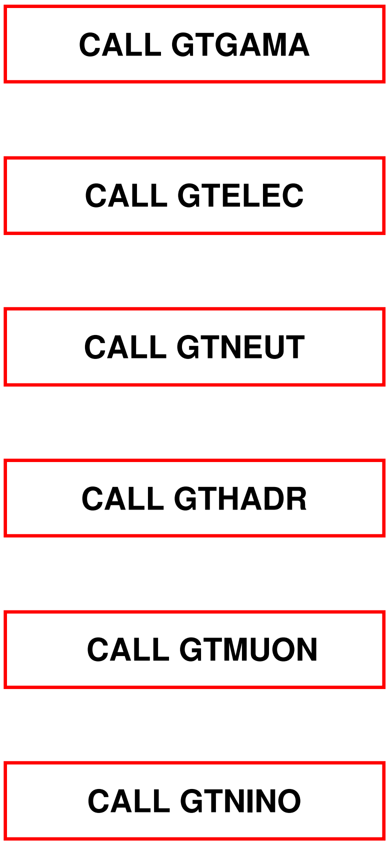


Figure 52: The traking routines

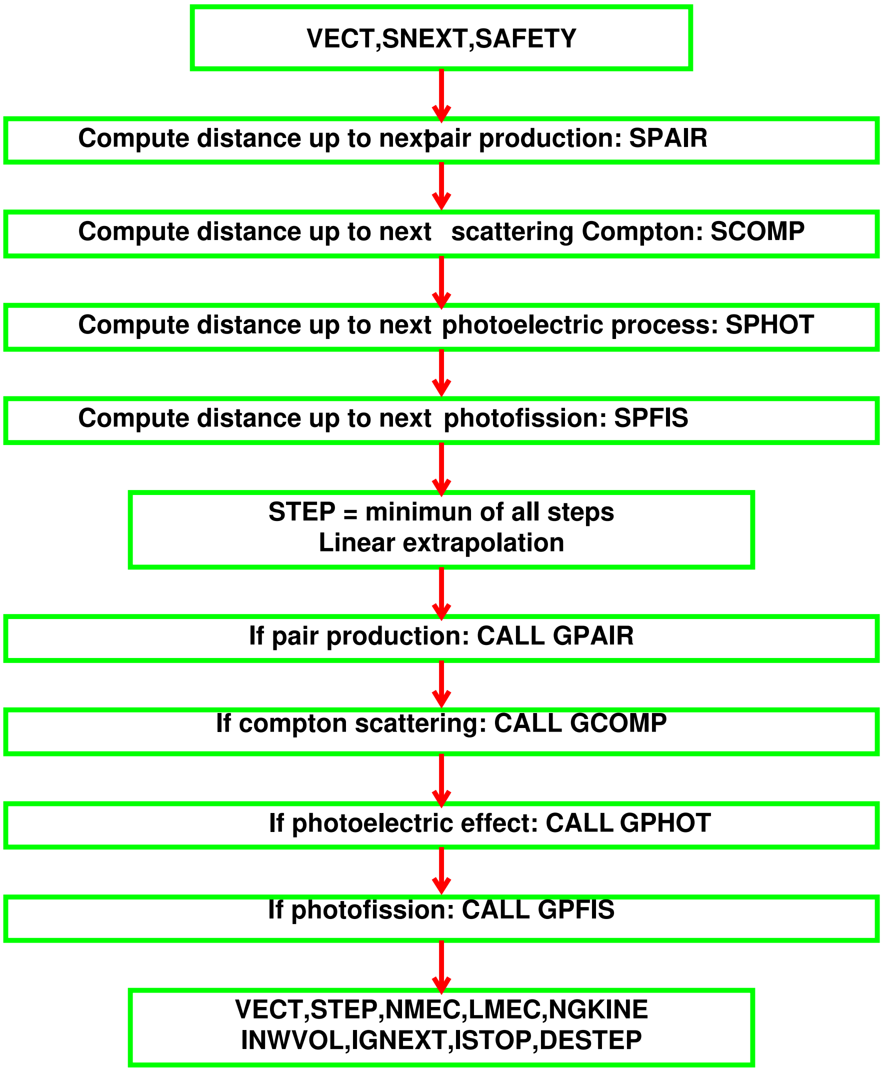


Figure 53: `GTGAMA` block diagram

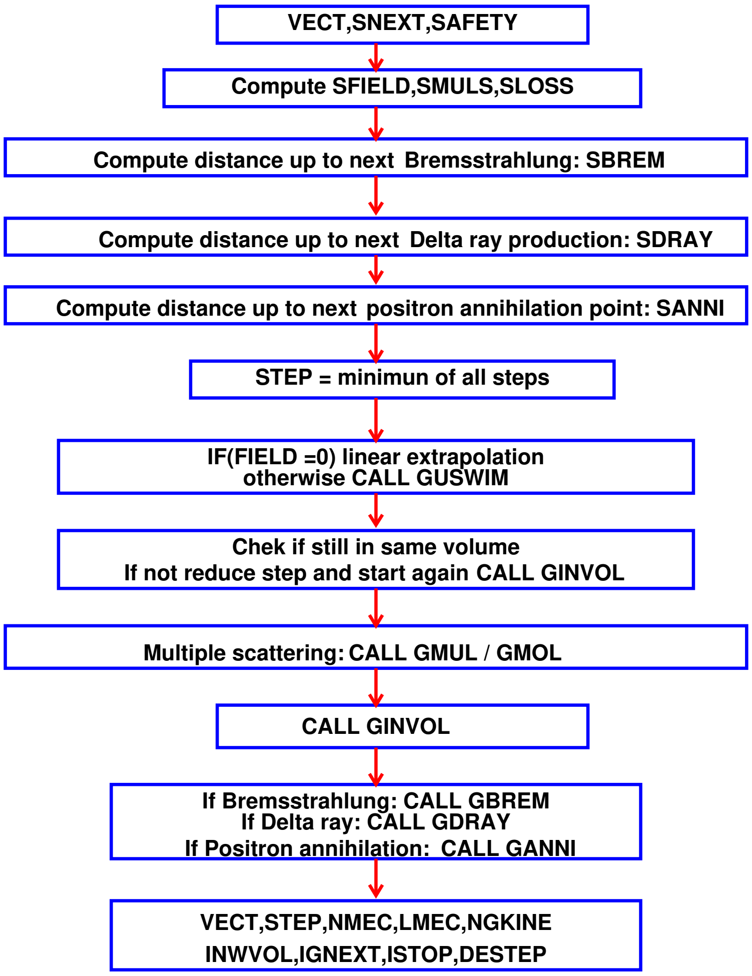


Figure 54: `GTELEC` block diagram

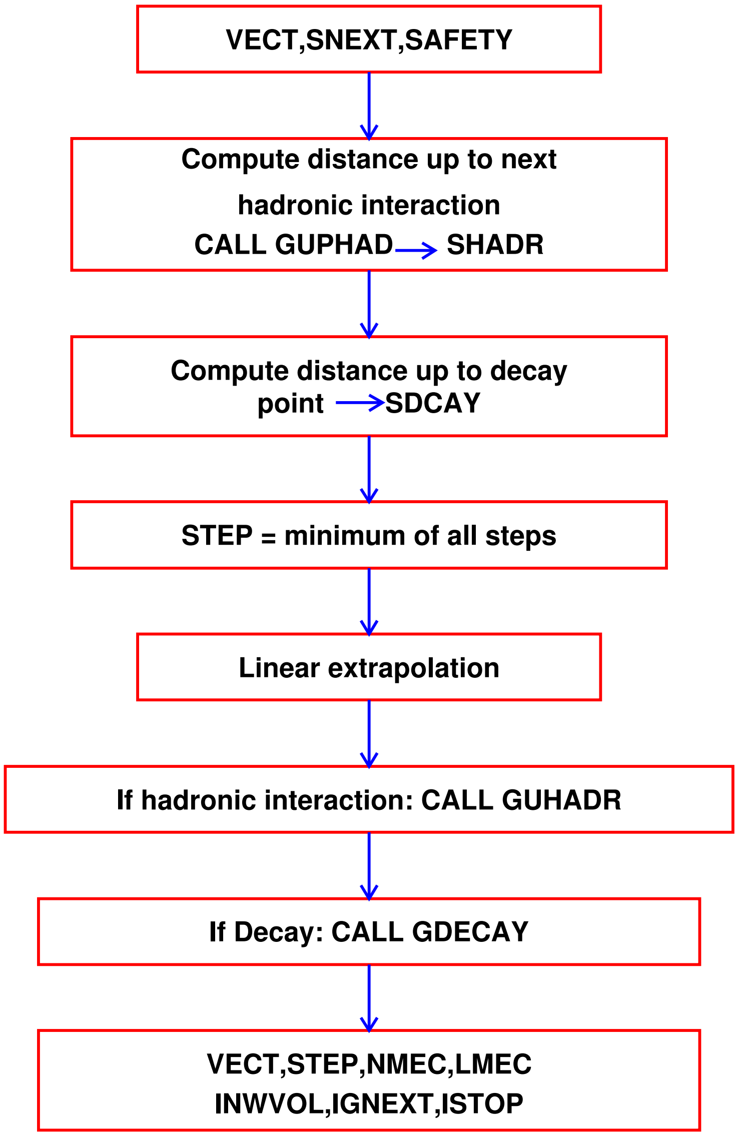


Figure 55: GTNEUT block diagram

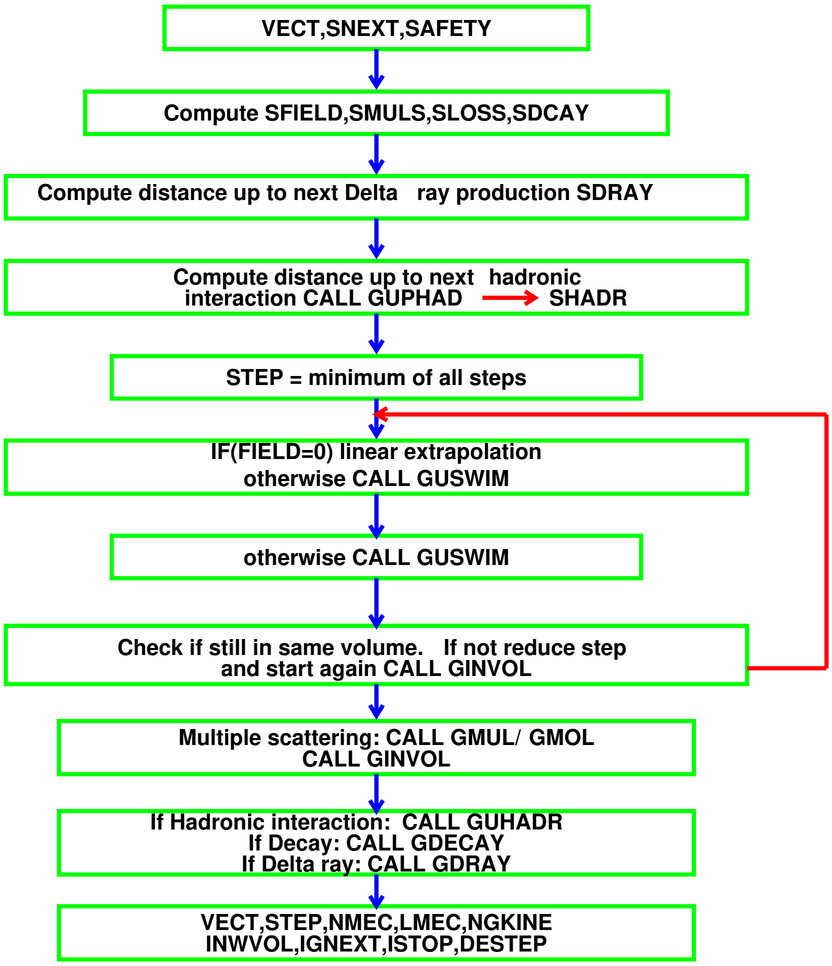


Figure 56: GTHADR block diagram

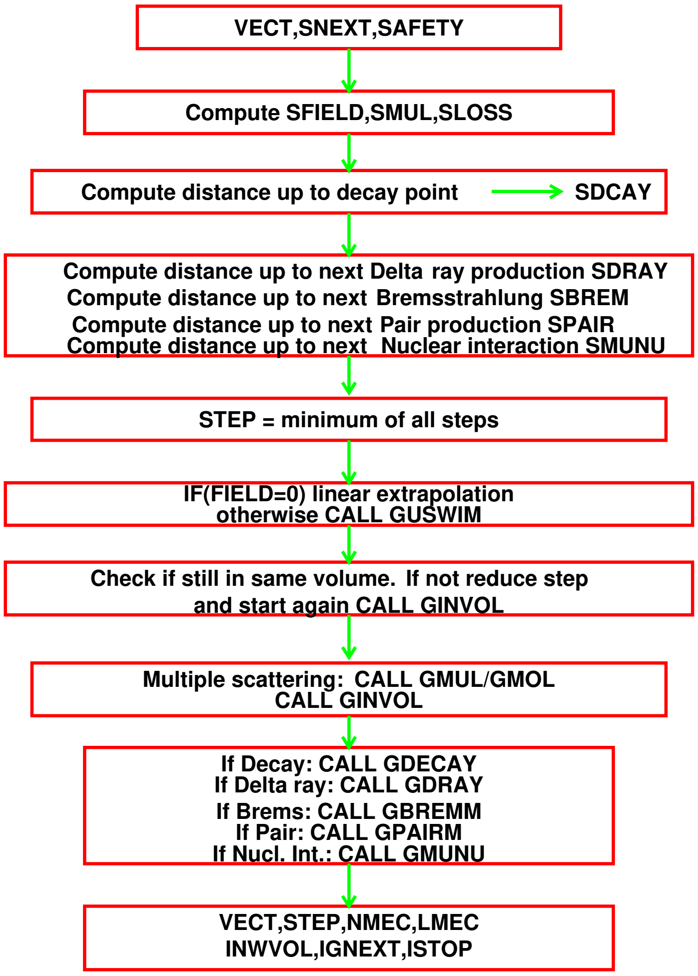


Figure 57: `GTMUON` block diagram

Origin : R.Brun **Submitted:** 01.06.83
Revision : **Revised:** 14.12.93
Documentation :

Storing secondary tracks in the stack

```
CALL GSKING (IT)
```

IT (INTEGER) number of the track to be stored for subsequent tracking, if IT=0 all secondary tracks generated in the current step are stored for tracking.

During transport, particles can generate new particles, called secondary. In GEANT these particles are stored in the common /GCKING/ (see [ZZZZ010] for an explanation of the variables in this common). The number of particles generated in the current step is NGKINE (common /GCKING/). If no user action is taken, the variable NGKINE is reset to 0 at the beginning of the subsequent step and these particles are not transported.

To be transported by GEANT, these particles need to be stored in the data structure JSTAK (see [TRACK399]) which normally is a LIFO stack: the last particle generated is the first one to be tracked when the current one stops. The order of tracking can be changed by the user, plase see [TRAK310] for more information on this subject.

The routine GSKING moves particles from the /GCKING/ common to the JSTAK structure, where GEANT looks for particles to transport.

When a particle is fetched from the JSTAK data structure, its place is freed for a new one, so the information on the initial kinematics of this track is lost. It may be useful in some occasion to record the initial kinematic of a track and save it till the end of the event. This is accomplished by storing the particle information both in the JSTAK data structure and in the JVERTX/JKINE one, which is permanent throughout the life of the event. As already said in [KINE100], this should not be done by the user calling directly the GSVERT/GSKINE routines during tracking, but rather controlling the action of GSKING via the array IFLGK in common /GCKING/. If the particle number IG generated in the current step ($1 \leq$ IG $\leq$ NGKINE) is being moved to the JSTAK *stack*, either singularly or together with all the other particles generated in the current step (setting the argument of GSKING to 0), the action performed depends on IFLGK(IG):

IFLGK(IG)<0 the particle is ignored;
=0 the particle is stored on the temporary stack JSTAK to be transported;
=1 the particle is stored on the temporary stack JSTAK to be transported *and* in the JKINE data structure attaching it to a vertex created in the current position (array VECT(I), I=1,3 in the /GCTRAK/ common);
>1 the particle is stored on the temporary stack JSTAK to be transported *and* in the JKINE data structure attaching it to vertex number IFLGK(IG), which must exist.

If IFLGK(IG)>0, after the call to GSKING, IFLGK(IG) is set to the newly created track number in JKINE. The number of the vertex used is returned in IFLGK(NGKINE+1). This feature allows the user to identify the created vertex and tracks via the routines GSVERU and GSKINU (see [KINE100]).

Example

In case of a hadronic interaction, discard the neutrinos, store the protons in the permanent data structure JVERTX/JKINE, and store all the other particles produced in the temporary stack.

Add to the protons in the JKINE bank the JVOLUM and copy number of the volume and the number of the tracking medium where they have been produced by an interaction.

```
      SUBROUTINE GUSTEP
+SEQ,GCKING,GCVOLU,GCTMED.
      CHARACTER*4 CHCASE
      DIMENSION UBUF(3)
      .
      .
      .
      CALL UHTOC(KCASE,4,CHCASE,4)
      IF(CHCASE.EQ.'HADR') THEN
         DO 10 IG=1, NGKINE
            IPART = GKIN(5,IG)
            IF(IPART.EQ.4) THEN
*---                Discard neutrinos
               IFLGK(IG)=-1
            ELSEIF(IPART.EQ.14) THEN
*---                Save protons in JKINE and transport them
               IFLGK(IG)=1
            ELSE
*---                Simply transport the rest
               IFLGK(IG)=0
            ENDIF
  10     CONTINUE
      ENDIF
*---                Perform action on all the particles
      CALL GSKING(0)
*---               Add information to the JKINE bank
      DO 20 IG=1,NGKINE
         IPART = GKIN(5,IG)
         IF(IPART.EQ.14) THEN
*---               This is a proton
*---               Get the track number
            IT = IFLGK(IG)
            NUBUF = 3
            UBUF(1) = LVOLUM(NLEVEL)
            UBUF(2) = NUMBER(NLEVEL)
            UBUF(3) = NUMED
            IADR = 0
*---               Store the information
            CALL GSKINU(IT,NUBUF,UBUF,IADR)
         ENDIF
  20  CONTINUE
```

Origin : F.Carminati **Submitted:** 14.12.93
Revision : **Revised:** 14.12.93
Documentation :

Altering the order of traking secondary particles

When secondary particles are produced during particle transport, if the user wants to transport them in turn, they must be stored by a call to `GSKING` or `GSKPHO` onto the temporary stack `JSTAK`. As explained in `[TRAK310]`, this is a LIFO structure, where the last particle introduced is the first to be extracted to be transported. It may be sometimes interesting to alter the order of tracking, for instance to decide whether the current event is worth completing or whether it should be skipped to go to next one.

It is possible in `GEANT` to alter the behaviour of the `JSTAK` structure using one of the words of the stack, normally reserved for the user. Stack ordering is activated via the data record `SORD 1`. This will set to 1 the variable `ISTORD` in common `/GCSTAK/`, which is by default initialised to 0.

When stack ordering is activated, the routine `GLTRAC`, which fetches the next particle to be tracked from `JSTAK`, instead of the last one, will select the one with the maximum value of the user word. This user word must be set by the user in the variable `UPWGHT` in the common `/GCTRAK/` prior to the call to `GSKING` to store each secondary.

If, for instance, all the protons must be tracked first, in order of production, and all the other particles after, again according to their time of production, the following code should appear in `GUSTEP`:

```
      SUBROUTINE GUSTEP
+SEQ,GCTRAK,GCKING
      .
      .
      .
      DO 10 IG=1,NGKINE
*---         Younger particles tracked first
*---         Add to the current time of flight the time delay of
*---         the secondary
         UPWGHT = -TOFG-TOFD(IG)
         IPART = GKIN(5,IG)
         IF(IPART.EQ.14) THEN
*---         This is a proton, add a 100 microseconds to have it
*---         tracked first
            UPWGHT = UPWGHT+1E-4
         ENDIF
*---         Now store the particle in the GEANT stack
         CALL GSKING(IG)
  10  CONTINUE
```

Users should be aware of the fact that ordering the stack will change the results, because transport depends on random numbers which will be different for the same particle in the two cases.

Origin : R.Brun **Submitted:** 01.11.83
Revision : F.Carminati **Revised:** 14.12.93
Documentation :

The temporary stack data structure JSTAK

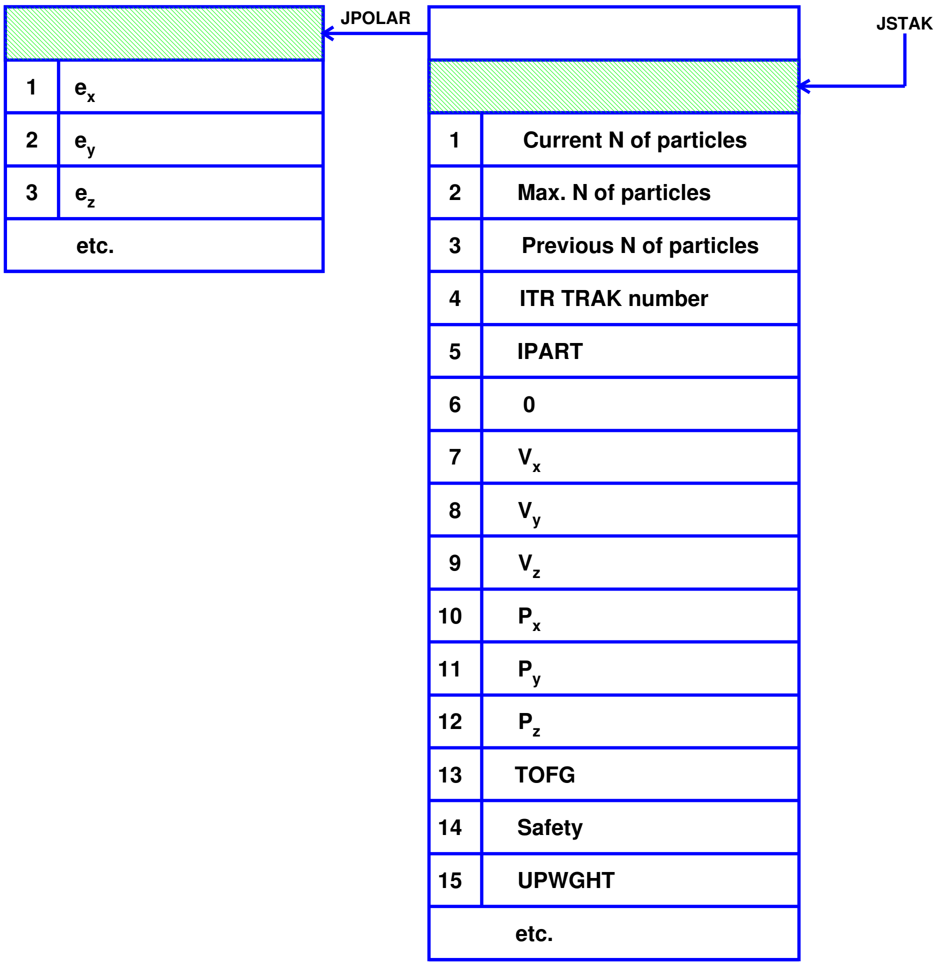


Figure 58: Layout of the temporary stack data structure JSTAK

The JSTAK bank is filled by GSKING.

Origin : R.Brun
Revision :
Documentation :

Submitted: 01.06.83
Revised: 15.12.93

Handling of track space points

`CALL` **GSXYZ**

Stores the position of the current particle from common `/GCTRAK/` into the data structure `JXYZ`.

`CALL` **GPJXYZ** `(NUMB)`

`NUMB` (`INTEGER`) track number, all tracks if =0;

Prints space points stored in the data structure `JXYZ` for track number `NUMB`.

`CALL` **GPCXYZ**

Prints tracking and physics parameters after the current step. This routine can be called from `GUSTEP`.

Some of these routines are called by `GDEBUG`, see `[BASE400]` for more information.

Origin : R.Brun
Revision :
Documentation :

Submitted: 01.11.83
Revised: 15.12.93

The space point data structure JXYZ

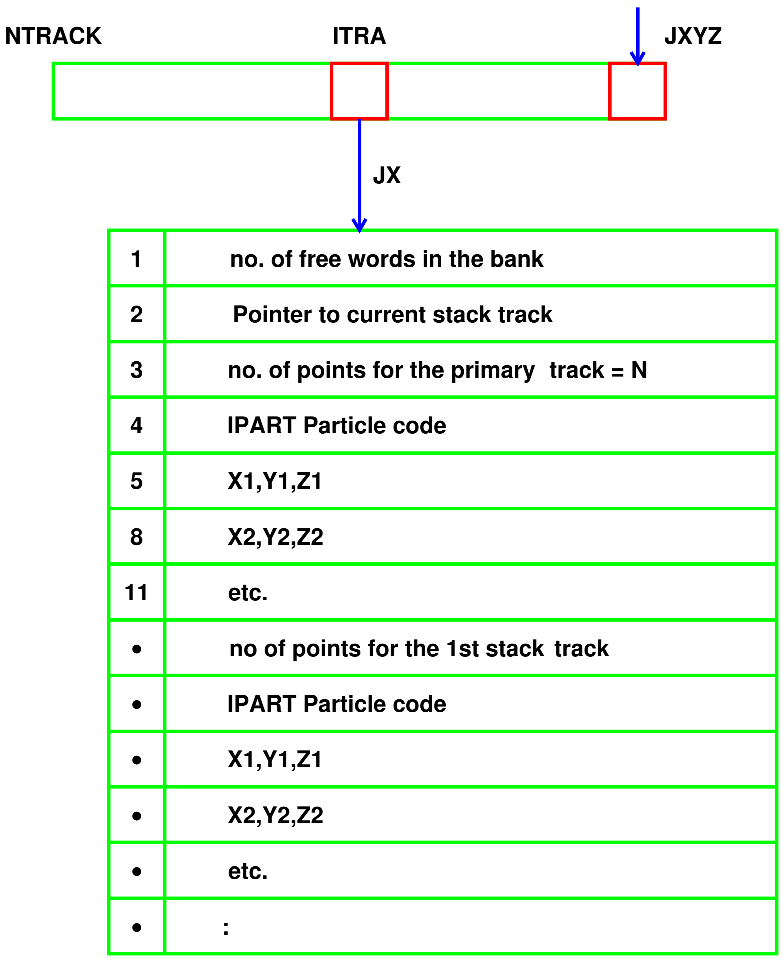


Figure 59: Layout of the JXYZ data structure

JX=LQ(JXYZ-ITRA) is the pointer to the space points of 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 from GUSTEP. The drawing routine GDXYZ gets the space coordinates from JXYZ.

Origin : R.Brun, M.Hansroul
Revision :
Documentation :

Submitted: 01.10.81
Revised: 16.12.93

Tracking routines in magnetic field

CALL **GUSWIM** (CHARGE,STEP,VECT,VOUT*)

CHARGE (REAL) particle charge in e unit;
STEP (REAL) step size in cm;
VECT (REAL) array of 7 with the initial coordinates, direction cosines and momentum (x, y, z, p_x/p, p_y/p, p_z/p, p);
VOUT (REAL) array of 7 with the final coordinates, direction cosines and momentum after the step.

This is the steering routine for calculating the deviation of a charged particle in magnetic field. It is called by the GEANT tracking routines: GTELEC, GTHADR, GTMUON and GTHION. Even if this routine follows the naming conventions of the user routines, the version provided in the GEANT library is quite adequate for most situation, and users should have a very good reason to modify it.

It calls GRKUTA, GHELIX, GHELX3 according to the value 1, 2 or 3 of IFIELD in common /GCTMED/. IFIELD correspond to the argument to the GSTMED routine (see [CONS200]). IFIELD = -1 is reserved for user decision in case GUSWIM is modified by the user.

CALL **GHELX3** (FIELD,STEP,VECT,VOUT*)

FIELD (REAL) field value in kilo-Gauss multiplied by the charge of the particle in e units;
STEP (REAL) step size in cm;
VECT (REAL) array of 7, same as for GUSWIM;
VOUT (REAL) array of 7, same as for GUSWIM;

Transport the particle by a length STEP in a magnetic field. The field is assumed to be uniform and parallel to the z axis. The value of the field is the FIELDM argument to the GSTMED routine (see [CONS200]). The particle is transported along an helix. The call to this routine is selected by GUSWIM when IFIELD=3 in the call to GSTMED.

CALL **GHELIX** (CHARGE,STEP,VECT,VOUT*)

CHARGE (REAL) particle charge in e units;
STEP (REAL) step size in cm;
VECT (REAL) array of 7, same as for GUSWIM;
VOUT (REAL) array of 7, same as for GUSWIM;

Transport the particle by a length STEP in a magnetic field. The magnetic field is calculated in the middle of the step (along a straight line) and it is supposed to be constant along the step and the particle is transported along an helix. The call to this routine is selected by GUSWIM when IFIELD=2 in the call to GSTMED. This routine is intended for magnetic fields with a small gradient. The value of the field is obtained from the routine GUFLD which must be coded by the user.

CALL **GRKUTA** (CHARGE,STEP,VECT,VOUT*)

The arguments have the same meaning than those of GHELIX. Transport a particle in magnetic field using Runge-Kutta method for solving the kinematic equations (Nystroem algorithm [1], procedure 25.5.20). The call to this routine is selected by GUSWIM when IFIELD=1 in the call to GSTMED. This method is the slower of the three and it must be used for magnetic fields with strong gradient. Again the magnetic field is obtained from the `GUFLD` routine which the user has to code.

```
CALL GUFLD ((VECT,F*)
```

`VECT` (`REAL`) array of 3 with the position in space where the field has to be returned;
`F` (`REAL`) array of 3 with the magnetic field in `VECT`, in kilo-Gauss units.

User routine to return the three components of the magnetic field at point `VECT` expressed in kilo-Gauss. This routine must be provided by the user for all tracking media where `IFIELD` is 1 or 2. This routine will be called very often, at least three times at each step, so all the care must be put in its optimisation.

TRAK Bibliography

[1] Handbook. Technical report, National Bureau of Standards.

Origin :
Revision :
Documentation : R.Brun

Submitted: 15.06.84
Revised: 19.12.93

The interactive version of GEANT

The interactive version is an essential tool for users who are in charge of the design of a detector. 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 KUIP [1] command processor. The user has to know a minimum about KUIP and should read at least the first chapter of the manual.

A set of encapsulated Postscript files have been collected in a new section XINT010. They illustrate the visualization possibilities of the interactive version of GEANT.

1 Invoking the interactive version

Instead of writing a MAIN program to initialize the Geant batch version, it is possible to use the MAIN interactive program (GXINT321.F) provided in /cern/pro/lib exactly as the library LIBGEANT321.A. The user has simply to insert in his link file the file GXINT321.F as the main program, followed by the user object code and by the library LIBGEANT321.A.

Example:

```
PROGRAM GXINT
*
* GEANT main program. To link with the MOTIF user interface
* the routine GPAWPP(NWGEAN,NWPAW) should be called, whereas
* the routine GPAW(NWGEAN,NWPAW) gives access to the basic
* graphics version.
*
PARAMETER (NWGEAN=3000000,NWPAW=1000000)
COMMON/GCBANK/GEANT(NWGEAN)
COMMON/PAWC/PAW(NWPAW)
*
CALL GPAWPP(NWGEAN,NWPAW)
*
END
```

The user has to set the desired value of NWGEAN and NWPAW for the GEANT and PAW Zebra stores, and to call the desired initialization routine:

- GPAW to initialize, besides GEANT, also HIGZ and to include the full functionality of PAW;
- GPAWPP to initialize, besides GEANT and HIGZ, also the Motif version and to include the full functionality of Paw++;

- USER initialization routine, to do anything else (for example a UGINIT- like routine or a gxint315.f–like routine).

The interactive version, after the initialization, gives the control to the user at the prompt GEANT > ; then it is possible to type and execute commands (corresponding to batch routines) to edit the geometry, the materials or the tracking media at run time. It is also possible to execute commands to visualize the detectors, to set the kinematics and to run events. Again interactively, one can spy the histograms, change the kinematics, and run more events (visualizing the tracks and the hits, for example). The GXINT chapter contains in the following pages a full description of the available GEANT commands. See the PAW, KUIP, DZDOC, HIGZ manuals for a description of the relative commands executable from GEANT. All the commands are also documented by an on-line help. (Try to type HELP at the first GEANT > prompt). In the interactive version, a COMIS interface is also available: COMIS is a FORTRAN interpreter which allows:

- to edit at run time important routines like UGEOM, GUSTEP, GUKINE, etc.
- to 'execute' them from the interactive session, without having to recompile and relink, by typing commands like CALL UGEOM.F.

Of course the interpreter is slower than the compiled object code, then, since GEANT321, it is also possible to invoke the native compiler and to link dinamically to the executable the compiled routine (see the COMIS manual for further details).

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 `KUIP` environment.

The commands are listed in subsection 1 - 13:

1. General GEANT
2. Clipping commands GEANT/CVOL
3. Drawing commands GEANT/DRAWING
4. Graphics control commands GEANT/DRAWING
5. Geometry commands GEANT/GEOMETRY
6. Volume creation commands GEANT/CREATE
7. Control commands GEANT/CONTROL
8. ZEBRA/RZ commands GEANT/RZ
9. ZEBRA/FZ commands GEANT/FZ
10. Data structure commands GEANT/DZ
11. Scanning commands GEANT/SCAN
12. Physics parameter commands GEANT/PHYSICS
13. List commands GEANT/LISTS

2 The Motif Interface

The interactive version 3.21 contains an object oriented Motif-based user interface. It can be accessed specifying 'm' as workstation type. The full functionality of the X11 version remains available, while new Motif-specific features have been added.

3 The main ideas

The `GEANT` data structures are considered as KUIP browsable classes and their contents as 'objects' on which one can perform actions (the `GEANT` commands). According to the Motif conventions, an object can be selected clicking the left button of the mouse (when the cursor is on the icon representing that object).

Clicking then on the right button of the mouse, a menu of possible commands will appear (double clicking on the left button the first action of this list will be executed); the selected command will perform the relative actions on the selected object. Such actions (like drawing, for example) can be executed either directly or via the use of an automatically opened Motif panel. Objects drawn in the graphics window can be 'picked' as well (for example, volumes, tracks, hits); clicking the right button when the cursor is on the selected object, a menu of possible actions on that object is displayed. Users can finally define Motif panels containing buttons corresponding to the most frequently used commands. An on-line help is available for any specific subject.

4 The Geant++ Executive Window

It replaces the normal dialog window; it contains a Transcript Pad, where the text output of the executed commands is displayed, and an Input Pad, where the user can still type the desired commands in the old style.

The `Geant++` Main File Browser

On the left side it displays a list of the `GEANT` data structures, of the available commands, file, macros and Zebra divisions used. Selecting one of them, the full list of icons representing the objects of that class is shown in the main area of the browser. Proceeding as described before, it is possible to perform actions on the classes (like create a new object) or on the objects belonging to them. It is possible to create menus of commands just clicking on the string 'commands' at the top line of the browser.

The Geant++ Graphics Window

Any object to be drawn in the graphics window can be stored in the current picture file (automatically opened after each NEXT command) via a call to IGPID (see Higz manual). It can be afterwards 'picked' as described before. In the case of commands executed via the use of Motif panels, some input values can be set with a slider ranging in the specifed range for the relative variable; moving the slider (after having clicked on the right-hand 'activating box') the relative action is performed in the graphics window when releasing the button of the mouse; when in 'drag mode', the action is performed *while* moving the slider (keeping the left button pressed): especially when double buffering has been selected, this can be useful for real time manipulations.

5 An Example

Start your `GEANT321` executable module (linked with GXINT321 and Motif1.2);

type 'm' as workstation type;

click the left button of the mouse after positioning the cursor on the string VOLU in the browser;

click the left button of the mouse after positioning the cursor on any icon in the main area of the browser;

click now the right button of the mouse and keep it pressed;

move the mouse to select the action 'Tree' and release the button;

the drawing of the logical tree will be displayed in the graphics window;

position the cursor on the drawing of a box (containing a volume name) in the graphics window, click the right button and keep it pressed;

release the button selecting the action 'Dspec';

the command DSPEC for that volume will be executed in a separate window;

repeat the exercise selecting this time the action ‘Dspe3d’;

the DSPEC will be executed in the first window, the volume specifications will be printed in a separate window and a Motif panel will appear;

click the left button of the mouse positioning the cursor in the Motif panel on the ‘Value changed’ button, and select the DRAG option;

click now the left button on the ‘activating box’ on the right of the ‘Theta’ slider;

click on the ‘Theta’ slider and, keeping pressed the left button of the mouse, move it right-wards;

the drawing in the graphics window will rotate;

release the button and type ‘igset 2buf 1’ in the executive window;

restart moving the slider as before.

Origin : Same **Submitted:** 15.06.84
Revision : **Revised:** 20.03.94
Documentation :

Introduction to the Interactive version of GEANT

1 GEANT

GEANT specific commands.

2 GEANT/CVOL

Clipping commands. The hidden line removal technique is necessary to visualize properly very complex detectors. At the same time, it can be useful to visualize the inner elements of a detector in detail. For this purpose, the commands menu CVOL has been developed: these commands allow subtractions (via boolean operation) of given shapes from any part of the detector, therefore showing its inner contents. It is possible to clip each different volume by means of a different shape (BOX , TUBE, CONE, SPHE are available). If '*' is given as the name of the volume to be clipped, all volumes are clipped by the given shape. A volume can be clipped at most twice (even by different shapes); if a volume is explicitely clipped twice, the '*' will not act on it anymore. Giving '.' as the name of the volume to be clipped will reset the clipping.

2.1 BOX cnnv [xmin xmax ymin ymax zmin zmax]

CNNV	C	" Name of volume to be clipped " D='* '
XMIN	R	" Lower limit of the Shape X coordinate " D=-10000.
XMAX	R	" Upper limit of the Shape X coordinate " D=-9999.
YMIN	R	" Lower limit of the Shape Y coordinate " D=-10000.
YMAX	R	" Upper limit of the Shape Y coordinate " D=-9999.
ZMIN	R	" Lower limit of the Shape Z coordinate " D=-10000.
ZMAX	R	" Upper limit of the Shape Z coordinate " D=-9999.

This command performs a boolean subtraction between the volume CNVV and a box placed in the MARS according the values of the given coordinates. See also CVOL. The following commands will clip by a box, with a vertex at the origin, the volume specified by NAME (a valid string for the NAME of the volume can be found using the DTREE command).

```
EXAMPLE -
dopt hide on
satt * seen -2
draw NAME 40 40 0 10 10 .01 .01
next
box NAME 0 1000 0 1000 0 1000
draw NAME 40 40 0 10 10 .01 .01
box .
```

2.2 TUBE cnvv [rmax zdem xmed ymed zmed]

CNVV	C	" Name of volume to be clipped " D='* '

RMAX R “ External radius of tube ” D=0.1
ZDEM R “ Half length of tube axis ” D=0.1
XMED R “ Center X coordinate ” D=-10000.
YMED R “ Center Y coordinate ” D=-10000.
ZMED R “ Center Z coordinate ” D=-10000.

This command performs a boolean subtraction between the volume CNVV and a tube; the tube has the given parameters and is placed in the MARS according the given coordinates of its center. See also CVOL. The following commands will clip, by a tube, positioned according to the given parameters, the volume specified by NAME (a valid string for the NAME of the volume can be found using the DTREE command).

```
EXAMPLE -
dopt hide on
satt * seen -2
draw NAME 40 40 0 10 10 .01 .01
next
tube * 500 1000 500 0 0
draw NAME 40 40 0 10 10 .01 .01
box .
```

2.3 CONE cnvv [rmax1 rmax2 zdem xmed ymed zmed]

CNVV C “ Name of volume to be clipped ” D='* '
RMAX1 R “ Min external radius ” D=0.1
RMAX2 R “ Max external radius ” D=0.1
ZDEM R “ Half length of cone axis ” D=0.1
XMED R “ Center X coordinate ” D=-10000.
YMED R “ Center Y coordinate ” D=-10000.
ZMED R “ Center Z coordinate ” D=-10000.

This command performs a boolean subtraction between the volume CNVV and a cone; the cone has the given parameters and is placed in the MARS according to the given coordinates of its center. See also CVOL. The following commands will clip by a cone, positioned according the given parameters, the volume specified by NAME (a valid string for the NAME of the volume can be found using the DTREE command).

```
EXAMPLE -
dopt hide on
satt * seen -2
draw NAME 40 40 0 10 10 .01 .01
next
cone * 1 750 1000 0 0 1000
draw NAME 40 40 0 10 10 .01 .01
box .
```

2.4 SPHE cnvv [rmax xmed ymed zmed]

CNVV C “ Name of volume to be clipped ” D='* '
RMAX R “ External radius of sphere ” D=0.1
XMED R “ Center X coordinate ” D=-10000.

YMED	R	“ Center Y coordinate ”	D=-10000.
ZMED	R	“ Center Z coordinate ”	D=-10000.

This command performs a boolean subtraction between the volume CNVV and a sphere; the sphere has the given parameters and is placed in the MARS according to the given coordinates of its center. See also CVOL. The following commands clip by a sphere, positioned according to the given parameters, the volume specified by NAME (a valid string for the NAME of the volume can be found using the DTREE command). EXAMPLE -

```
dopt hide on
satt * seen -2
draw NAME 40 40 0 10 10 .01 .01
next
sphe * 500 0 0 500
draw NAME 40 40 0 10 10 .01 .01
box .
```

2.5 VALCUT xcut ycut zcut

XCUT	R	“x coordinate of cutted value”	D=0.
YCUT	R	“y coordinate of cutted value”	D=0.
ZCUT	R	“z coordinate of cutted value”	D=0.

It allows the cutting in the ray-tracing. All the volumes are cutted from XCUT to +BIG along the x axis, from YCUT to +BIG along the y axis and from ZCUT to +BIG along the z axis.

3 GEANT/DRAWING

Drawing commands. These commands allow the visualization in several ways of the volumes defined in the geometrical data structure. It is possible to draw the logical tree of volumes belonging to the detector (DTREE), to show their geometrical specification (DSPEC,DFSPC), to draw them and their cut views (DRAW, DCUT). Moreover, it is possible to execute these commands when the hidden line removal option is activated; in this case, the volumes can be also either translated in the space (SHIFT), or clipped by boolean operation (CVOL). In addition, it is possible to fill the surfaces of the volumes with solid colours when the shading option (SHAD) is activated. Several tools (ZOOM, LENS) have been developed to zoom detailed parts of the detectors or to scan physical events as well. Finally, the command MOVE will allow the rotation, translation and zooming on real time parts of the detectors or tracks and hits of a simulated event. Ray-tracing commands. In case the command (DOPT RAYT ON) is executed, the drawing is performed by the Geant ray-tracing; automatically, the color is assigned according to the tracking medium of each volume and the volumes with a density lower/equal than the air are considered transparent; if the option (USER) is set (ON) (again via the command (DOPT)), the user can set color and visibility for the desired volumes via the command (SATT), as usual, relatively to the attributes (COLO) and (SEEN). The resolution can be set via the command (SATT * FILL VALUE), where (VALUE) is the ratio between the number of pixels drawn and 20 (user coordinates). Parallel view and perspective view are possible (DOPT PROJ PARA/PERS); in the first case, we assume that the first mother volume of the tree is a box with dimensions 10000 X 10000 X 10000 cm and the view point (infinetely far) is 5000 cm far from the origin along the Z axis of the user coordinates; in the second case, the distance between the observer and the origin of the world reference system is set in cm by the command (PERSP NAME VALUE); grand-angle or telescopic effects can be achieved changing the scale factors in the command (DRAW). When the final picture does not occupy the full window, mapping the space before tracing can speed up the drawing, but can also produce less precise results; values from 1 to 4 are allowed in the command (DOPT MAPP VALUE), the mapping being more precise for increasing (VALUE); for (VALUE = 0) no mapping is performed (therefore max precision and lowest speed). The command (VALCUT) allows the cutting of the detector by three planes ortogonal to the x,y,z axis. The attribute (LSTY) can be set by the command SATT for any desired volume and can assume values from 0 to 7; it determines the different light processing to be performed for different materials: 0 = dark-matt, 1 = bright-matt, 2 = plastic, 3 = ceramic, 4 = rough-metals, 5 = shiny-metals, 6 = glass, 7 = mirror. The detector is assumed to be in the dark, the ambient light luminosity is 0.2 for each basic hue (the saturation is 0.9) and the observer is assumed to have a light source (therefore he will produce parallel light in the case of parallel view and point-like-source light in the case of perspective view).

3.1 DRAW name [theta phi psi u0 v0 su sv]

`NAME`	C	"Volume name"
`THETA`	R	"Viewing angle theta (for 3D projection)" `R=0.:180.`
`PHI`	R	"Viewing angle phi (for 3D projection)" `R=0.:360.`
`PSI`	R	"Viewing angle psi (for 2D rotation)" `R=0.:360.`
`U0`	R	"U-coord. (horizontal) of volume origin"
`V0`	R	"V-coord. (vertical) of volume origin"
`SU`	R	"Scale factor for U-coord."
`SV`	R	"Scale factor for V-coord."

```
CALL GDRAW(name,theta,phi,psi,u0,v0,su,sv)
```

If optional parameters are missing, the corresponding values are taken from the common /GCDRAW/. This command will draw the volumes, selected with their graphical attributes, set by the SATT facility. The drawing may be performed with hidden line removal and with shading effects according to the value of the options HIDE and SHAD; if the option SHAD is ON, the contour's edges can be drawn or not. If the option

HIDE is ON, the detector can be exploded (BOMB), clipped with different shapes (CVOL), and some of its parts can be shifted from their original position (SHIFT). When HIDE is ON, if the drawing requires more than the available memory, the program will evaluate and display the number of missing words (so that the user can increase the size of its ZEBRA store). Finally, at the end of each drawing (with HIDE on), the program will print messages about the memory used and statistics on the volumes' visibility. The following commands will produce the drawing of a green volume, specified by NAME, without using the hidden line removal technique, using the hidden line removal technique, with different linewidth and colour (red), with solid colour, with shading of surfaces, and without edges. Finally, some examples are given for the ray-tracing. (A possible string for the NAME of the volume can be found using the command DTREE).

```
EXAMPLE -
satt * seen -2
satt NAME colo 3
draw NAME 40 40 0 10 10 .01 .01
next
dopt hide on
draw NAME 40 40 0 10 10 .01 .01
next
satt NAME colo 2
satt NAME lwid 4
draw NAME 40 40 0 10 10 .01 .01
next
dopt shad on
satt * lwid 1
satt NAME fill 1
draw NAME 40 40 0 10 10 .01 .01
next
satt NAME fill 3
draw NAME 40 40 0 10 10 .01 .01
next
dopt edge off
draw NAME 40 40 0 10 10 .01 .01
dopt rayt on
satt * fill 20
dopt mapp 1
draw NAME 40 40 0 10 10 .01 .01
dopt proj pers
persp NAME 500
draw NAME 40 40 0 10 10 1 1
valcut 100 100 100
dopt mapp 0
dopt user on
satt NAM1 seen 0
satt NAM2 colo 2
draw NAME 40 40 0 10 10 5 5
```

3.2 SPOT xlpos ylpos zlpos inten

XLPOS	R	"x coordinate of light source"
YLPOS	R	"y coordinate of light source"
ZLPOS	R	"z coordinate of light source"
INTEN	I	"intensity of light source"

This point-like light source can be moved in the space and its intensity can be changed (INTEN going from 0 to 10) relatively to the ambience light.

3.3 VAR5D tseqto nproc nmptot totmby tseq tlat tnet

`TSEQTO`	R	"total sequential time"
`NPROC`	I	"number of processors"
`NMPTOT`	I	"number of message passing"
`TOTMBY`	R	"total megabytes transfert"
`TSEQ`	R	"not parallelized code"
`TLAT`	R	"latency time"
`TNET`	R	"network speed in Mbytes/sec"

It sets the values of the parameters expressed in the formula and specify which variables must be assumed as x,y,z (setting their value to 1001,1002,1003, respectively).

3.4 RANG5D x1min x1max y1min y1max z1min z1max

`X1MIN`	R	"x coordinate min"
`X1MAX`	R	"x coordinate max"
`Y1MIN`	R	"y coordinate min"
`Y1MAX`	R	"y coordinate max"
`Z1MIN`	R	"z coordinate min"
`Z1MAX`	R	"z coordinate max"

It sets the range for the x,y,z variables.

3.5 DVOLUME n namnum chnrs [theta phi psi u0 v0 su sv]

`N`	I	"Number of elements in arrays LNAMES and LNUMBS" `D=1`
`NAMNUM`	C	"Volume names and numbers (ex. `"`NAME1,NR1,NAME2,NR2`"`)"
`CHNRS`	C	"Reference system used" `D='MARS'`
`THETA`	R	"Viewing angle theta (for 3D projection)" `R=0.:360.`
`PHI`	R	"Viewing angle phi (for 3D projection)" `R=0.:360.`
`PSI`	R	"Viewing angle psi (for 2D rotation)" `R=0.:180.`
`U0`	R	"U-coord. (horizontal) of volume origin"
`V0`	R	"V-coord. (vertical) of volume origin"
`SU`	R	"Scale factor for U-coord."
`SV`	R	"Scale factor for V-coord."

Possible `CHNRS` values are:

`MARS`

`DRS`

```
CALL GDRVOL(n,lnames,lnumbs,nrs,theta,phi,psi,u0,v0,su,sv)
```

N is the number of levels from the top of the geometry structure to the volume lnames(n),lnumbs(n) to be drawn. NAMNUM contain the arrays lnames and lnumbs, identifying the path, in pairs and separated by commas; for example (with n=2) : 'lname(1),lnumbs(1),lname(2),lnumbs(2) ' CHNRS is the name of the reference system used: MARS for MAster Reference System or DRS for Daughter Reference System. NRS=0 for MARS or NRS⟨⟩0 for DRS If optional parameters are missing, the current values in /GCDRAW/ are taken.

3.6 DCUT name caxis cutval [u0 v0 su sv]

`NAME`	C	"Volume name"
`CAXIS`	C	"Axis value"
`CUTVAL`	R	"Cut plane distance from the origin along the axis"
`U0`	R	"U-coord. (horizontal) of volume origin"
`V0`	R	"V-coord. (vertical) of volume origin"
`SU`	R	"Scale factor for U-coord."
`SV`	R	"Scale factor for V-coord."

Possible `CAXIS` values are:

`X`

`Y`

`Z`

```
CALL GDRAWC(name,iaxis,cutval,u0,v0,su,sv)
```

The cut plane is normal to caxis (X,Y,Z), corresponding 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. When HIDE Mode is ON, it is possible to get the same effect with the CVOL/BOX command.

3.7 DXCUT name cutthe cutphi cutval [theta phi u0 v0 su sv]

`NAME`	C	"Volume name"
`CUTTHE`	R	"Theta angle of the line normal to cut plane" `R=0.:360.`
`CUTPHI`	R	"Phi angle of the line normal to cut plane" `R=0.:360.`
`CUTVAL`	R	"Cut plane distance from the origin along the axis"
`THETA`	R	"Viewing angle theta (for 3D projection)" `R=0.:360.`
`PHI`	R	"Viewing angle phi (for 3D projection)" `R=0.:360.`
`U0`	R	"U-coord. (horizontal) of volume origin"
`V0`	R	"V-coord. (vertical) of volume origin"
`SU`	R	"Scale factor for U-coord."
`SV`	R	"Scale factor for V-coord."

```
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 and 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.

3.8 SHIFT cnvn xxxx yyyy zzzz

CNVN	C	" Name of volume to be shifted " D='*'
XXXX	R	" Shift along X axis " D=0.
YYYY	R	" Shift along Y axis " D=0.
ZZZZ	R	" Shift along Z axis " D=0.

To draw a volume shifted from its initial position when hidden line removal is ON. It can be useful if you want to extract a volume or some volumes from the detector to show them more clearly. The last requested SHIFT for each volume NAME is performed. Moreover, the SHIFT of each volume will be performed starting from where its mother has been shifted, so that it's easier to SHIFT nicely sets of volumes using the mother-daughter relationships. If '.' is given as the name of the volume to be shifted, the shifts for all volumes will be reset. The following commands will produce the translation along the Z-axis of the previously drawn volume:

```
EXAMPLE -
dopt hide on
satt * seen -2
draw NAME 40 40 0 10 10 .01 .01
shift NAME 0 0 10
```

3.9 BOMB boom

BOOM	R	" Exploding factor for volumes position " D=0. R=-10.:10.

To 'explode' the detector. If BOOM is positive (values smaller than 1. are suggested, but any value is possible) all the volumes are shifted by a distance proportional to BOOM along the direction between their centre and the origin of the MARS; the volumes which are symmetric with respect to this origin are simply not shown. BOOM equal to 0 resets the normal mode. A negative (greater than -1.) value of BOOM will cause an 'implosion'; for even lower values of BOOM the volumes' positions will be reflected respect to the origin. This command can be useful to improve the 3D effect for very complex detectors. The following commands will make explode the detector:

```
EXAMPLE -
dopt hide on
satt * seen 1
draw NAME 40 40 0 10 10 .01 .01
bomb 1
next
draw NAME 40 40 0 10 10 .01 .01
```

3.10 DTREE [name levmax iselt]

NAME	C	"Volume name" D='␣'
LEVMAX	I	"Depth level" D=3 R=-15:15
ISELT	I	"Options " D=111

This command allows the drawing of the logical tree, displaying the name, the multiplicity and other information about the volumes, via a call to GDTREE(name,levmax,isel): if the third parameter is not given (default), the command will produce the drawing of the tree displaying, for each volume, the number of the following levels (red arrows) and of the preceeding levels (green arrows); then the control is automatically given to the mouse: clicking on the left button when the cursor is inside a volume's pave will perform a

DSPEC for that volume; doing the same when the cursor is on a red arrow, will perform a DTREE for the relative volume (the number of levels displayed depending on the clicked arrow); doing the same for the 'i-th' green arrow of a given volume, will perform a DTREE for its mother-volume staying 'i' levels before. If running with X-windows, the drawing of the specification (DSPEC) is performed in a different window to speed up the scanning of the tree. Iterating this procedure it is possible to analyse very easily and quickly any kind of tree. Clicking the right button of the mouse will return the control to the command mode. If the ISELT parameter is given, then the TREE will work as in the previous version, with ISELT up to 10001. The following command will perform a drawing of the tree and give the control to the user via the mouse:

```
EXAMPLE -
dtree NAME 3
```

3.11 DSPEC name

`NAME`	C	"Volume name"

Trough a call to GDSPEC(name), this command allows one to show three views of the volume (two cut-views and a 3D view), together with its geometrical specifications. The 3D drawing will be performed according the current values of the options HIDE and SHAD and according the current CVOL clipping parameters for that volume.

3.12 D3DSPEC name [teta3 phi3 psi3 u03 v03 zm3]

`NAME`	C	"Volume name"
`TETA3`	R	"Theta angle" `D=40. R=0.:180.`
`PHI3`	R	"Phi angle" `D=40. R=0.:360.`
`PSI3`	R	"Psi angle" `D=0. R=0.:360.`
`U03`	R	"U-coord. (horizontal) of volume origin" `D=10. R=-40.:40.`
`V03`	R	"V-coord. (vertical) of volume origin" `D=10. R=-40.:40.`
`ZM3`	R	"Zoom factor for current size factors" `D=1. R=0.00001:10.`

Trough a call to GSPE3D, this command allows one to show the volume (3D views in real time), together with its geometrical specifications (if using MOTIF). The 3D drawing will be performed according the current values of the options HIDE and SHAD and according the current CVOL clipping parameters for that volume.

3.13 DFSPC name [csort cinter]

`NAME`	C	"Volume name"
`CSORT`	C	"Alphabetic sorting flag" `D='N'`
`CINTER`	C	"Interactive/Batch version" `D='I'`

Possible `CSORT` values are:

`Y`

`N`

`0`

`1`

Possible `CINTER` values are:

I

B

0

1

```
CALL GDFSPC(name,isort,inter)
```

Same as DSPEC, but it will draw the specifications for all the volumes. If the alphabetic sorting flag is YES, all pictures will be drawn in ascending alphabetic order; isort is set to 1. If INTERACTIVE, (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.

3.14 DTEXT x0 y0 text size angle lwid cent

X0	R	"X-coord. (horizontal) of text string" D=10. R=0.:20.
Y0	R	"Y-coord. (vertical) of text string" D=10. R=0.:20.
TEXT	C	"Text string" D='GEANT'
SIZE	R	"Character size (cm)" D=.5
ANGLE	R	"Rotation angle (deg)" D=0. R=0.:360.
LWID	I	"Line width" D=4
CENT	C	"Centering option" D='CENT'

Possible CENT values are:

CENT

LEFT

RIGH

```
CALL GDRAWT(x0,y0,text,size,angle,lwid,opt)
```

It allows one to draw some text in the current picture. Now more than 160 colours are available. The text colour must be set via the command IGSET. The size of the text will follow the zooming factors in the view banks.

3.15 DVECTOR xvect yvect npoint

XVECT	C	"Vector containing X-coord. (horizontal)"
YVECT	C	"Vector containing Y-coord. (vertical)"
NPOINT	I	"Number of coord."

Draw a polyline of 'npoint' point via a call to GDRAWV(xvect,yvect,npoint) where xvect and yvect are two KUIP vectors

3.16 DSCALE u v

U	R	"U-coord. (horizontal) of the centre of scale"
V	R	"V-coord. (vertical) of the centre of scale"

```
CALL GDSCAL(u,v)
```

It draws a scale centered in U,V.

3.17 DAXIS x0 y0 z0 dx

X0	R	"X-coord. of axis origin"
Y0	R	"Y-coord. of axis origin"
Z0	R	"Z-coord. of axis origin"
DX	R	"Axis size"

```
CALL GDAXIS(x0,y0,z0,dx)
```

This commmand superimposes the axis of the MARS on the current picture. It is useful for finding immediately the orientation of the current drawing of the detector in the space.

3.18 DMAN u v type

U	R	"U-coord. (horizontal) of the centre of man"
V	R	"V-coord. (vertical) of the centre of man"
TYPE	C	"Man, Wm1, Wm2, Wm3" D='MAN'

Possible TYPE values are:

MAN

WM1

WM2

WM3

```
CALL GDMAN(u,v),CALL GDWMN1(u,v),CALL GDWMN2(u,v),CALL GDWMN2(u,v)
```

It superimposes the picure of a man or of a woman, chosen among three different ones, with the same scale factors as the detector in the current drawing.

3.19 DHEAD [isel name chrsiz]

ISEL	I	"Option flag" D=111110
NAME	C	"Title" D='␣'
CHRSIZ	R	"Character size (cm) of title NAME" D=0.6

```
CALL GDHEAD(isel,name,chrsiz)
```

ISEL =

```
0      to have only the header lines
xxxxx1 to add the text name centered on top of header
xxxx1x to add global detector name (first volume) on left
xxx1xx to add date on right
xx1xxx to select thick characters for text on top of header
```

```
x1xxxx to add the text 'EVENT NR x' on top of header
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 and CHRSIZ the character size in cm of text name.

3.20 MEASURE

Position the cursor on the first point (u1,v1) and hit the space bar(GKS). Position the cursor on the second point (u2,v2) and hit the space bar(GKS). Clicking the left button of the mouse (X11) will have the same effect as hiting the space bar (GKS). The command will compute and print the distance in space separating the two points on the projection view. It can be useful to measure distances either between volumes or between tracks or hits.

3.21 PICK

Activates graphic input to identify detector elements in a cut view. Clicking on the left button of the mouse when the cursor is in a given point of the drawing and clicking again (outside the detector) will produce the following effect: a line joininig the two points will be drawn together with the name and the medium number of the volume picked with the first clicking close to the second point.

3.22 MOVE name [nopt]

`NAME`	C	"Volume name"	`D=' '`
`NOPT`	C	"S=sample mode,T=tracks,H=hits"	`D=' '`

Positioning some daughter volumes inside a 'mother', it can be important to check if overlaps between such volumes have occurred. Instead of putting the drawing in a view bank, zooming, and iterating the process for different viewing angles of the same detector, the MOVE facility has been developed (for machines running with X11): it is sufficient to draw a view of the volumes to be analysed (after setting the proper SEEN, COLO, etc. attributes) and then to enter 'MOVE' followed by the same 'NAME' used for the last command DRAW. The detector will appear in a panel with five buttons at the bottom: THETA, PHI, TRASL, ZOOM, OFF. Clicking on the left button of the mouse, when the cursor is inside the THETA area, will rotate the detector along the polar angle theta according to the backward-to-forward movement of the mouse (clicking up and down the left button if not in sample mode); clicking on the right button of the mouse will stop the rotation; clicking now on the left button of the mouse when inside the PHI area will activate a rotation along the polar angle phi. In the same way, activating the TRASL button, the detector can be translated in the u,v plane of the screen according to the 2D-movement of the mouse. Finally, activating the ZOOM button, the detector will be zoomed (or unzoomed) according to the backward-to-forward movement of the mouse. Clicking on the OFF button will return the control to the 'command mode'. The MOVE command will work also with hidden line removal and shading options (when SHAD is on the background will be black); moreover, if the volumes are clipped, exploded, shifted, etc., they will be 'MOVED' with these features as well. Tracks and hits of a previously stored physical event can be moved together with the detector, allowing a dynamical 3-D analysis of the simulated events. Clicking the central button of the mouse when a good view of the event is found, will stop any movement and the mouse will allow the normal picking capabilities first for the tracks and then for the hits. After clicking of the right button, the normal movement will restart to find another interesting view of the event and to iterate the process. The MOVE is also available in sample mode. The following commands will produce a drawing of a volume and then will give the control to the MOVE panel; try the following possibilities:

```
EXAMPLE 1 -
dopt hide off
satt * seen -2
```

```
draw NAME 40 40 0 10 10 .01 .01
move NAME
EXAMPLE 2 -
dopt hide on
satt * seen -2
draw NAME 40 40 0 10 10 .01 .01
move NAME
EXAMPLE 3 -
dopt shad on
satt * colo 3
satt * fill 2
dopt edge off
draw NAME 40 40 0 10 10 .01 .01
move NAME
```

3.23 MOVE3D name [theta phi psi u0 v0 su sv sz nopt]

NAME	C	"Volume name" D=' '
THETA	R	"Viewing angle theta (for 3D projection)" D=40. R=0.:180.
PHI	R	"Viewing angle phi (for 3D projection)" D=40. R=0.:360.
PSI	R	"Viewing angle psi (for 2D rotation)" D=0. R=0.:180.
U0	R	"U-coord. (horizontal) of volume origin" D=10. R=0.:20.
V0	R	"V-coord. (vertical) of volume origin" D=10. R=0.:20.
SU	R	"Scale factor for U-coord." D=0.01
SV	R	"Scale factor for V-coord." D=0.01
SZ	R	"Scale zoom factor" D=1. R=0.1:10.
NOPT	C	"T=tracks,H=hits" D=' '

Possible NOPT values are:

T

H

Same functionality of the command MOVE interfaced with MOTIF.

3.24 PERSP name distt [samp]

NAME	C	"Volume name" D=' '
DISTT	R	"Volume distance from observer" D=1000.
SAMP	C	"Control to the mouse" D='OFF '

To control the perspective according to the variation of the distance between the observer and the object (if PROJ has the value PERS). If SAMP is ON the control of the distance is given via the mouse.

3.25 LENS knum [ksam]

KNUM	I	"View bank identifier" D=1
KSAM	C	"Sample mode " D='OFF '

Interactive zooming for detectors and events when running with X-windows. Using this command, when showing the contents of a view bank, it is possible to click (left button) in two points of the drawing (which will represent the left upper corner and the right bottom corner of the part to be zoomed). After the second click a new 'window' will appear to fit the frame defined by the two clicks and it will show a zoomed view as seen from a lens with those dimensions. Clicking now the central button will translate the lens over the drawing, while clicking the right button will stop it. Moreover, clicking the left button of the mouse, the lens will increase (or decrease) its magnification power according to the backward-to-forward movement of the mouse. A click on the right button will stop this action and it is possible to restart the translation of the lens or, clicking on the right button again, to make the lens disappear. It is then possible to open another 'window-lens' with different dimensions. Thus, this command can be useful to scan detailed parts of a detector or to scan hits and showers for events. Clicking the right button when no lens is displayed will return the control to the 'command mode'. The LENS is also available in sample mode when KSAM is 'ON'. The following commands will fill a view bank and will allow to scan the detector and an event previously stored via the use of LENS (when running with X-windows):

```
EXAMPLE -
satt * seen 1
dopen 1
draw NAME 40 40 0 10 10 .01 .01
dxyz 0
dhits * * 0 0 .2
dclose
dsh 1
lens 1 on
```

3.26 ZOOM [zfu zfv isel uz0 vz0 u0 v0]

ZFU	R	"Zoom factor for U-coord. (horizontal)"	D=2.
ZFV	R	"Zoom factor for V-coord. (vertical)"	D=2.
ISEL	I	"Options"	D=1
UZ0	R	"U-coord. of the centre of zoom rectangle"	D=10. R=0.:20.
VZ0	R	"V-coord. of the centre of zoom rectangle"	D=10. R=0.:20.
U0	R	"U-coord. of the centre of resulting zoomed rectangle"	D=10. R=0.:20.
V0	R	"V-coord. of the centre of resulting zoomed rectangle"	D=10. R=0.:20.

```
CALL GDZOOM(zfu,zfv,uz0,vz0,u0,v0)
```

This command sets the zoom parameters that will be used by subsequent 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. The use of the space bar is replaced by the left button of the mouse running with X11: If isel=0 :

```
1. position the cursor at (uz0,vz0)
2. type the space bar (GKS)
```

(u0,v0 are chosen at centre of screen) If isel=1 :

```
1. position the cursor at first corner of zoom rectangle
2. type the space bar (GKS)
3. position the cursor at second corner of zoom rectangle
4. type the space bar (GKS)
```

(zfu,zfv are chosen according to the zoom rectangle; uz0,vz0 are chosen at the centre of the zoom rectangle; u0,v0 are chosen at centre of screen) If isel=2 :

```
1. position the cursor at (uz0,vz0)
2. type the space bar (GKS)
3. position the cursor at (u0,v0)
4. type the space bar (GKS)
```

If isel=1000+n and running with X-windows:

```
1. n must be the identifier of an active view bank
2. clicking on the left button of the mouse will display
   a zoomed view (computed around the cursor position) of
   the previous drawing in a new window
3. it is now possible to iterate the zooming from the new window
4. clicking on the right button will return the control to the
   main window
5. clicking on the left button it is possible to open new windows
   zooming in other points of the detector
6. clicking on the right button when the main window is active
   will return the control to the 'command mode'.
```

3.27 DXYZ [itra]

`ITRA` I "Track number" `D=0`

```
CALL GDXYZ(itra)
```

Draw tracks previously stored via GSXYZ.

3.28 KXYZ [epsilo]

`EPSILO` R "Delta angle" `D=0.25`

```
CALL GKXYZ(epsilo)
```

The picking of track points requires the JXYZ data structure and is repeated until the character typed is 'Q' or 'q' (GKS) or the right button of the mouse is clicked (X11). EPSILO is the delta angle used for picking; if EPSILO=0 there is no optimization performed and over all the track points the one nearest to the pick point is taken.

3.29 DPART [itra isel size]

`ITRA` I "Track number" `D=0`
`ISEL` I "Option flag" `D=11`
`SIZE` R "Character size (cm) for particle names" `D=0.25`

```
CALL GDPART(itra,isel,size)
isel=x1 to draw the track number
isel=1x to draw the particle name
```

3.30 DHITS [chuset chudet itra isymb ssymb]

CHUSET	C	"User set identifier" `D='*'`
CHUDET	C	"User detector identifier" `D='*'`
ITRA	I	"Number of the selected track" `D=0`
ISYMB	I	"Character selection number" `D=0`
SSYMB	R	"Size of characters (cm)" `D=0.1`

CALL GDHITS(chuset,chudet,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. The hit colour will follow the value of TXCI (text colour) for isymb⟩0, the value of PMCI (polymarkers colour) for isymb⟨0, the value of PLCI (polyline colour) for isymb=0.

3.31 KHITS [chuset chudet epsilo]

CHUSET	C	"User set identifier" `D='*'`
CHUDET	C	"User detector identifier" `D='*'`
EPSILO	R	"Pick aperture" `D=0.1`

```
CALL GKHITS(chuset,chudet,epsilo)
```

The picking of hit points requires the appropriate JSET data structure have been filled and is repeated until the character typed is 'Q' or 'q' (GKS) or the right button of the mouse is clicked (X11). If the character typed to pick is 'K' or 'k' then the kinematics of the corresponding track is also printed. The search is made of all the hits of all tracks in detector CHUDET of set CHUSET. 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.

3.32 DCHIT [chuset chudet itra isymb sizmax ihit hitmin hitmax]

CHUSET	C	"User set identifier" `D='*'`
CHUDET	C	"User detector identifier" `D='*'`
ITRA	I	"Number of the selected track" `D=0`
ISYMB	I	"Character selection number" `D=0`
SIZMAX	R	"Maximum character size (cm)" `D=1`
IHIT	I	"Index of array HITS" `D=4`
HITMIN	R	"Lower boundary of HITS(IHIT)" `D=0`
HITMAX	R	"Upper boundary of HITS(IHIT)" `D=0`

```
CALL GDCHIT(chuset,chudet,itra,isymb,sizmax,ihit,hitmin,hitmax)
```

The character plotted at each hit point may be chosen via CSYMB; isymb is composed as:

```
     -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), the array containing the calorimeter quantity, with HITMIN and HITMAX defining its range. The maximum character size (used in overflow) is SIZMAX.

```
SIZE = SIZMAX * ( HITS(IHIT) - HITMIN ) / HITMAX
```

3.33 DUVIEW name type cpxtyp [iview]

NAME	C	"Detector name"	
TYPE	C	"View name"	
CPXTYP	C	"Complexity name"	
IVIEW	I	"View number where picture is stored"	D=0

```
CALL GUVIEW(name,type,cpxtyp,iview)
```

4 GEANT/GRAPHICS_CONTROL

Graphics control commands.

4.1 DOPEN iview

`IVIEW` I "View number"

```
CALL GDOPEN(iview)
```

When a drawing is very complex and requires a long time to be executed, it can be useful to store it in a view bank: after a call to DOPEN and the execution of the drawing (nothing will appear on the screen), and after a necessary call to DCLOSE, the contents of the bank can be displayed in a very fast way through a call to DSHOW; therefore, the detector can be easily zoomed many times in different ways. Please note that the pictures with solid colours can now be stored in a view bank or in 'PICTURE FILES'.

4.2 DSHOW [iview]

`IVIEW` I "View number"

```
CALL GDSHOW(iview)
```

It shows on the screen the contents of a view bank. It can be called after a view bank has been closed.

4.3 DELETE iview

`IVIEW` I "View number"

```
CALL GDELET(iview)
```

It deletes a view bank from memory.

4.4 DCLOSE

```
CALL GDCLOS
```

It closes the currently open view bank; it must be called after the end of the drawing to be stored.

4.5 CHANGEWK

CALL GCHNWK It open a new workstation (if not already opened) and activate it (deactivating the default one).

4.6 RESETWK

CALL GRESWK It deactivate the previosly activated workstation and reactivate the default one.

4.7 SSETAT iopt

`IOPT` C "Attribute name"

Set current attribute.

4.8 SSETVA [rval]

`RVAL` R "Attribute value" `D=1. R=-10.:10.`

Set current attribute value.

4.9 SATT [name iopt ival]

`NAME` C "Volume name" `D='* '`
`IOPT` C "Name of the attribute to be set" `D='DEFA'`
`IVAL` I "Value to which the attribute is to be set" `D=10000`

```
CALL GSATT(name,iopt,ival)
```

name='*' stands for all the volumes. iopt can be chosen among the following :

```
'WORK'   0=volume name is inactive for the tracking
         1=volume name is active for the tracking (default)
'SEEN'   0=volume name is invisible
         1=volume name is visible (default)
        -1=volume invisible with all its descendants in the tree
        -2=volume visible but not its descendants in the tree
'LSTY'   line style 1,2,3,... (default=1)
         LSTY=7 will produce a very precise approximation for
         revolution bodies.
'LWID'   line width -7,...,1,2,3,..7 (default=1)
         LWID<0 will act as abs(LWID) was set for the volume
         and for all the levels below it. When SHAD is 'ON', LWID
         represent the linewidth of the scan lines filling the surfaces
         (whereas the FILL value represent their number). Therefore
         tuning this parameter will help to obtain the desired
         quality/performance ratio.
'COLO'   colour code -166,...,1,2,..166 (default=1)
         n=1=black
         n=2=red;    n=17+m, m=0,25, increasing luminosity according to
'm';
         n=3=green;  n=67+m, m=0,25, increasing luminosity according to
'm';
         n=4=blue;   n=117+m, m=0,25, increasing luminosity according to
'm';
         n=5=yellow; n=42+m, m=0,25, increasing luminosity according to
'm';
         n=6=violet; n=142+m, m=0,25, increasing luminosity according to
'm';
         n=7=lightblue; n=92+m, m=0,25, increasing luminosity according to
'm';
         colour=n*10+m, m=1,2,...9, will produce the same colour
         as 'n', but with increasing luminosity according to 'm';
         COLO<0 will act as if abs(COLO) was set for the volume
```

```
         and for all the levels below it.
         When for a volume the attribute FILL is > 1 (and the
         option SHAD is on), the ABS of its colour code must be < 8
         because an automatic shading of its faces will be
         performed.
'FILL'   (1992) fill area  -7,...,0,1,...7 (default=0)
         when option SHAD is 'on' the FILL attribute of any
         volume can be set different from 0 (normal drawing);
         if it is set to 1, the faces of such volume will be filled
         with solid colours; if ABS(FILL) is > 1, then a light
         source is placed along the observer line, and the faces of
         such volumes will be painted by colours whose luminosity
         will depend on the amount of light reflected;
         if ABS(FILL) = 1, then it is possible to use all the 166
         colours of the colour table, becouse the automatic shading
         is not performed;
         for increasing values of FILL the drawing will be performed
         with higher and higher resolution improving the quality (the
         number of scan lines used to fill the faces increases with FILL);
         it is possible to set different values of FILL
         for different volumes, in order to optimize at the same time
         the performance and the quality of the picture;
         FILL<0 will act as if abs(FILL) was set for the volume
         and for all the levels below it.
         This kind of drawing can be saved in 'picture files'
         or in view banks.
         0=drawing without fill area
         1=faces filled with solid colours and resolution = 6
         2=lowest resolution (very fast)
         3=default resolution
         4=................
         5=................
         6=................
         7=max resolution
         Finally, if a coloured background is desired, the FILL
         attribute for the first volume of the tree must be set
         equal to -abs(colo), colo being >0 and <166.
'SET '   set number associated to volume name
'DET '   detector number associated to volume name
'DTYP'   detector type (1,2)
```

4.10 SCALE gscu gscv

`GSCU`	R	"Scale factor for U-coord."
`GSCV`	R	"Scale factor for V-coord."

Change the scale factors GSCU and GSCV in /GCDRAW/.

4.11 COLOR icol

`ICOL` I "Colour code" `D=1`

```
CALL GDCOL(-abs(icol))
```

4.12 LWID lwidth

`LWIDTH` I "Line width code" `D=1`

```
CALL GDLW(-abs(lwidth))
```

4.13 NEXT

Clear screen (start a new picture on graphics file, if opened).

4.14 DOPT [iopt ival]

`IOPT` C "Option name" `D='*'`
`IVAL` C "Option value" `D='*'`

```
CALL GDOPT(iopt,ival)
```

To set/modify the drawing options.

```
IOPT   IVAL      Action
THRZ    ON       Draw tracks in R vs Z
        OFF (D)  Draw tracks in X,Y,Z
        180
        360
PROJ    PARA (D) Parallel projection
        PERS     Perspective
TRAK    LINE (D) Trajectory drawn with lines
        POIN       " " with markers
HIDE    ON       Hidden line removal using the CG package
        OFF (D)  No hidden line removal
SHAD    ON       Fill area and shading of surfaces.
        OFF (D)  Normal hidden line removal.
RAYT    ON       Ray-tracing on.
        OFF (D)  Ray-tracing off.
EDGE    OFF      Does not draw contours when shad is on.
        ON  (D)  Normal shading.
MAPP    1,2,3,4  Mapping before ray-tracing.
        0   (D)  No mapping.
USER    ON       User graphics options in the raytracing.
        OFF (D)  Automatic graphics options.
```

4.15 SIZE [xsize ysize]

`XSIZE` R "Size along X" `D=20.`
`YSIZE` R "Size along Y" `D=20.`

Set the size of the picture. On the terminal, the pictures will have the ratio YSIZE/XSIZE, and, if a metafile is produced, pictures will be YSIZE by XSIZE cm. This command sets the parameters for the normalisation transformation number 1 to [0-XSIZE], [0-YSIZE].

4.16 SPERS dpers

DPERS	R	"Distance from the origin"

Set the variable dpers in /GCDRAW/, representing the distance from the origin when using option PERSpective.

4.17 MAP_COLOR [icadd icval]

ICADD	I	"Colour table index"	D=0
ICVAL	I	"Colour table value"	D=0

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.

5 GEANT/GEOMETRY

Geometry commands.

5.1 OPTI iopti

`IOPTI`	C	"GSORD optimisation level" `D='0' Minus`

Possible `IOPTI` values are:

`-1`

`2`

This flag controls the tracking optimisation performed via the GSORD routine:

```
1 no optimisation at all; GSORD calls disabled;
0 no optimisation; only user calls to GSORD kept;
1 all non-GSORDered volumes are ordered along the best axis;
2 all volumes are ordered along the best axis.
```

5.2 SVOLU name shape numed npar par

`NAME`	C	"Volume name"
`SHAPE`	C	"Volume type"
`NUMED`	I	"Tracking medium number"
`NPAR`	I	"Number of shape parameters"
`PAR`	C	"Vector containing shape parameters"

```
CALL GSVOLU(name,shape,numed,par,npar,ivolu)
```

where par is a KUIP vector. It creates a new volume in the JVOLUM data structure.

5.3 SPOS name number mother x0 y0 z0 irot only

`NAME`	C	"Volume name"
`NUMBER`	I	"Copy number of the volume"
`MOTHER`	C	"Mother volume name"
`X0`	R	"X coord. of the volume in mother ref. sys."
`Y0`	R	"Y coord. of the volume in mother ref. sys."
`Z0`	R	"Z coord. of the volume in mother ref. sys."
`IROT`	I	"Rotation matrix number w.r.t. mother ref. sys."
`ONLY`	C	"ONLY/MANY flag"

```
CALL GSPOS(name,number,mother,x0,y0,z0,irot,only)
```

It positions a previously defined volume in the mother.

5.4 SDVN name mother ndiv caxis

`NAME`	C	"Volume name"

MOTHER	C	"Mother volume name"
NDIV	I	"Number of divisions"
CAXIS	C	"Axis value"

Possible CAXIS values are:

X

Y

Z

1

2

3

```
CALL GSDVN(name,mother,ndiv,iaxis)
```

X,Y,Z of CAXIS will be translated to 1,2,3 for IAXIS. It divides a previously defined volume.

5.5 PVOLU numb

NUMB	I	"Volume ID"

```
CALL GPVOLU(numb)
```

Prints volumes' specifications.

5.6 SROTM irot theta1 phi1 theta2 phi2 theta3 phi3

IROT	I	"Rotation matrix number"
THETA1	R	"Polar angle for axis I" D=0. R=0.:180.
PHI1	R	"Azimuthal angle for axis I" D=0. R=0.:360.
THETA2	R	"Polar angle for axis II" D=0. R=0.:180.
PHI2	R	"Azimuthal angle for axis II" D=0. R=0.:360.
THETA3	R	"Polar angle for axis III" D=0. R=0.:180.
PHI3	R	"Azimuthal angle for axis III" D=0. R=0.:360.

```
CALL GSROTM(irot,theta1,phi1,theta2,phi2,theta3,phi3)
```

It defines the rotation matrix number IROT.

5.7 PROTM numb

NUMB	I	"Matrix ID"

```
CALL GPROTM(numb)
```

Print matrixes' specifications.

5.8 STMED ntmed name nmat isvol ifield fieldm tmaxfd stemax deemax epsil stmin

NTMED	I	"Tracking medium number" D=1
NAME	C	"Tracking medium name"
NMAT	I	"Material number" D=1
ISVOL	I	"Sensitive volume flag" D=0
IFIELD	I	"Magnetic field" D=0
FIELDM	R	"Max. field value (Kilogauss)" D=0
TMAXFD	R	"Max. angle due to field (deg/step)" D=0.01
STEMAX	R	"Max. step allowed" D=1.E+10
DEEMAX	R	"Max. fraction of energy lost in a step" D=0.01
EPSIL	R	"Tracking precision (cm)" D=0.01
STMIN	R	"Min. step due to continuos processes (cm)" D=0.1

```
      CALL GSTMED(ntmed,name,nmat,isvol,ifield,fieldm,tmaxfd,
     +            stemax,deemax,epsil,stmin,0,0)
```

IFIELD = 0 if no magnetic field; IFIELD = -1 if user decision in GUSWIM; IFIELD = 1 if tracking performed with GRKUTA; IFIELD = 2 if tracking performed with GHELIX; IFIELD = 3 if tracking performed with GHELX3.

5.9 PTMED numb

NUMB	I	"Medium ID"

```
CALL GPTMED(numb)
```

Print tracking media's specifications.

5.10 EDITV [isel name]

ISEL	I	"Options" D=0
NAME	C	"Volume name" D=' '

```
CALL GEDITV(isel,name)
```

When the routine prompts for input parameters that do not need to be changed, type return. ISEL is used to select the editing operation to be performed:

```
ISEL=0, CALL GGCLOS
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
```

5.11 CADINT fname aname nbins lunit lunit2 inst site dept resp

FNAME	C	"Name of the SET file" D='example.set'
ANAME	C	"Name of the volume"
NBINS	I	"Number of the instances" D=1
LUNIT	I	"Logical unit number for SET file" D=66
LUNIT2	I	"Logical unit number for material file" D=67
INST	C	"Name of your institute" D='CERN'
SITE	C	"Name of site" D='MEYRIN'
DEPT	C	"Name of departement" D='CN'
RESP	C	"Name of sender" D='god_knows_who'

```
CALL GTXSET(fname,aname,nbins,lunit,lunit2,inst,site,dept,resp)
```

This command produces a SET file describing the given volume with the contents currently set visible. (Use the visibility attribute, see SATT SEEN.) The description is given as a flat assembly related to the global coordinate system. The ouput can be read into CAD systems (EUCLID-IS) trough a SET interface. A list of materials of the volumes in the SET file and the GEANT tree is written into a file with the same filename as the SET file, but with extension .mat.

5.12 REUCLID lun fname

LUN	I	"Logical unit of the file to be read" R=1:100
FNAME	C	"Name of the EUCLID file to be read"

```
CALL GREUCL(LUN,FNAME)
```

Calls the routine to read into GEANT a geometry from an ASCII file written by the EUCLID-GEANT interface.

5.13 WEUCLID lun fname topvol [number nlevel]

LUN	I	"Logical unit of the file to be written" R=1:100
FNAME	C	"Name of the EUCLID file to be written"
TOPVOL	C	"Volume name of the starting node"
NUMBER	I	"Copy number of TOPVOL (relevant for GSPOSP)" D=1
NLEVEL	I	"Number of levels in the tree structure" D=15

```
CALL GWEUCL(LUN,FNAME)
```

Calls the routine to write the current GEANT geometry into an ASCII file in EUCLID compatible format.

6 GEANT/CREATE

It creates volumes of the given shape interactively. CALL GSVOLU(name,shape,numed,par,npar,ivolu) where par is a KUIP vector

6.1 SBOX name numed halfx halfy halfz [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
HALFX	R	"Half X length"
HALFY	R	"Half Y length"
HALFZ	R	"Half Z length"
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.2 STRD1 name numed hlfdwx hlfupx halfy halfz [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
HLFDWX	R	"Half X length in Lower Z Surface"
HLFUPX	R	"Half X length in Upper Z Surface"
HALFY	R	"Half Y length"
HALFZ	R	"Half Z length"
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.3 STRD2 name numed hlfdwx hlfupx hlfdwy hlfupy halfz [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
HLFDWX	R	"Half X length in Lower Z Surface"
HLFUPX	R	"Half X length in Upper Z Surface"
HLFDWY	R	"Half Y length in Lower Z Surface"
HLFUPY	R	"Half Y length in Upper Z Surface"
HALFZ	R	"Half Z length"
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.4 STUBE name numed inrad outrad halfz [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
INRAD	R	"Inside Radius"
OUTRAD	R	"Outside Radius"
HALFZ	R	"Half Z length"
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.5 STUBS name numed inrad outrad halfz sphi ephi [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
INRAD	R	"Inside Radius"
OUTRAD	R	"Outside Radius"
HALFZ	R	"Half Z length"
SPHI	R	"Start of section PHI" R=0.:360.
EPHI	R	"End of section PHI" R=0.:360.
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.6 SCONE name numed inrdw outrdw inrup outrup halfz [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
INRDW	R	"Inside Radius in Lower Z Surface"
OUTRDW	R	"Outside Radius in Lower Z Surface"
INRUP	R	"Inside Radius in Upper Z Surface"
OUTRUP	R	"Outside Radius in Upper Z Surface"
HALFZ	R	"Half Z length"
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.7 SCONS name numed inrdw outrdw inrup outrup halfz sphi ephi [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
INRDW	R	"Inside Radius in Lower Z Surface"
OUTRDW	R	"Outside Radius in Lower Z Surface"
INRUP	R	"Inside Radius in Upper Z Surface"
OUTRUP	R	"Outside Radius in Upper Z Surface"
HALFZ	R	"Half Z length"
SPHI	R	"Start of section PHI" R=0.:360.
EPHI	R	"End of section PHI" R=0.:360.
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.8 SSPHE name numed inrad outrad sphi ephi stheta etheta [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
INRAD	R	"Inside Radius"
OUTRAD	R	"Outside Radius"
SPHI	R	"Start of section PHI" R=0.:360.
EPHI	R	"End of section PHI" R=0.:360.
STHETA	R	"Start of section THETA"
ETHETA	R	"End of section THETA"
YESNO	C	"GSPOSP option" D='NO'

Possible YESNO values are:

YES

NO

6.9 SPARA name numed halfx halfy halfz axis phi theta [yesno]

NAME	C	"Volume name"
NUMED	I	"Tracking medium number"
HALFX	R	"Half X length"
HALFY	R	"Half Y length"
HALFZ	R	"Half Z length"
AXIS	R	"Angle of Y mid-faces segment to Y axis" R=0.:360.
PHI	R	"PHI angle of Low Z mid-face to High Z mid-face segment" R=0.:360.

`THETA` R "THETA angle of mid-low-Z-face to mid-high-Z-face segment" `R=0.:360.`

`YESNO` C "GSPOSP option" `D='NO'`

Possible `YESNO` values are:

`YES`

`NO`

7 GEANT/CONTROL

Control commands.

7.1 KINE ikine [pkine1 pkine2 pkine3 pkine4 pkine5 pkine6 pkine7 pkine8 pkine9 pkine10]

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

Set the variables in /GCFLAG/ IKINE, PKINE(10)

7.2 RUNG idrun idevt

IDRUN	I	"User run number"
IDEVT	I	"User starting event number"

Set the run number and the starting value for the user event number.

7.3 SORD istord

ISTORD	C	"Flag to control user ordering of the stack" D='1'

Possible ISTORD values are:

1

0

If ISTORD is set to 1, the particle with the highest value of the user weight UPWGHT will be selected to be tracked next.

7.4 GTIME timint timend itime

TIMINT	R	"Total time after initialisation"
TIMEND	R	"Time reserved for the termination phase"
ITIME	I	"Frequency of control printing"

These commands have limited use in the interactive version. In particular the value of TIMINT is disregarded by GEANT.

7.5 TRACK

Restart tracking, clearing the track and hit banks, but keeping the kinematics.

7.6 TRIGGER [n]

`N` I "Number of events" `D=1`

Start one or more new events.

7.7 RNDM [iseed1 iseed2]

`ISEED1` I "First seed for the random number generator"
`ISEED2` I "Second seed for the random number generator"

Set the seeds for the random number generator. If no numbers are given, the currents seeds are printed.

7.8 SWITCH iswi ival

`ISWI` I "Switch number"
`IVAL` I "New switch value"

Change one element of array ISWIT(10) in /GCFLAG/

7.9 MZLOGL level

`LEVEL` I "MZ log level" `D=0`

Set the log level for the MZ package of ZEBRA: CALL MZLOGL(0,level)

```
LEVEL = -3   no messages at all
        -2   error messages only
        -1   terse logging
         0   normal
        +1   log rare events
        +2   log calls to MZ routines
```

7.10 PRINT name number

`NAME` C "Name"
`NUMBER` I "Number" `D=0`

```
CALL GPRINT(name,number)
```

7.11 OUTPUT_LP lout

`LOUT` I "New output unit"

To change lout in /GCUNIT/ Note: unit numbers 5,11,12,13,14,15 are reserved and cannot be used.

7.12 PHITS [chuset chudet numhi]

CHUSET	C	"User set" D='*'
CHUDET	C	"User detector" D='*'
NUMHI	I	"Hit number" D=0

```
CALL GPHITS(chuset,chudet)
```

7.13 PDIGI [chuset chudet]

CHUSET	C	"User set" D='*'
CHUDET	C	"User detector" D='*'

```
CALL GPDIGI(chuset,chudet)
```

7.14 SMATE imat namate a z dens radl absl ubuf nwbuf

IMAT	I	"Material number"
NAMATE	C	"Material name"
A	R	"Atomic weight"
Z	R	"Atomic number"
DENS	R	"Density"
RADL	R	"Radiation lenght"
ABSL	R	"Absorption lenght"
UBUF	R	" "
NWBUF	I	" "

```
CALL GSMATE(imat,namate,a,z,dens,radl,absl,ubuf,nwbuf)
```

7.15 SMIXT imat namate a z dens nlmat wmat

IMAT	I	"Material number"
NAMATE	C	"Material name"
A	R	"Atomic weight"
Z	R	"Atomic number"
DENS	R	"Density"
NLMAT	I	"Flag for WMAT"
WMAT	R	"Relative weights or n. of atoms in molecule"

```
CALL GSMIXT(imat,namate,a,z,dens,nlmat,wmat)
```

7.16 PMATE numb

NUMB	I	"Material number"

```
CALL GPMATE(numb)
```

7.17 PRMAT imate ipart mecan

`IMATE`	I	"Material number"
`IPART`	I	"Particle number"
`MECAN`	C	"Mechanism"

```
CALL GPRMAT(imate,ipart,mecan,nekbin,elow)
```

7.18 PLMAT imate ipart mecan [idm]

`IMATE`	I	"Material number"
`IPART`	I	"Particle number"
`MECAN`	C	"Mechanism"
`IDM`	I	"ID mode option" `D=0`

CALL GPLMAT(imate,ipart,mecan,nekbin,elow,idm)

```
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)
```

If MECAN = 'ALL' all the mechanisms are histogrammed. If the material number is negative, the cross sections relative to material ABS(IMATE) will be histogrammed in barns rather than in 1/cm.

7.19 DRMAT imate ipart [mecan]

`IMATE`	I	"Material number"
`IPART`	I	"Particle number"
`MECAN`	C	"List of mechanism" `D='ALL'`

CALL GDRMAT(imate,ipart,mecan,nmec) If MECAN = 'ALL' all the mechanisms are plotted. If the material number is negative, the cross sections relative to material ABS(IMATE) will be plotted in barns rather than in 1/cm. Note that it is not possible to plot anything if GSTMED has not been called for the material number IMATE.

7.20 STPAR itmed chpar parval

`ITMED`	I	"Medium number"
`CHPAR`	C	"Cut or mechanism"
`PARVAL`	R	"Value"

CALL GSTPAR(itmed,chpar,parval)

7.21 SPART ipart napart itrtyp amass charge tlife ubuf nwbuf bratio mode

`IPART`	I	"Particle number"
`NAPART`	C	"Particle name"

ITRTYP	I	“ ”
AMASS	R	“Mass”
CHARGE	R	“Charge”
TLIFE	R	“Lifetime”
UBUF	R	“ ”
NWBUF	I	“ ”
BRATIO	R	“Branching ratios”
MODE	I	“Decay mode”

CALL GSPART(ipart,napart,itrtyp,amass,charge,tlife,ubuf,nwbuf); CALL GSDK(ipart,bratio,mode)

7.22 PPART numb

NUMB	I	“Particle number”

CALL GPPART(numb)

7.23 PRKINE numb

NUMB	I	“Track number”

CALL GPKINE(numb)

7.24 DEBUG [ideb]

IDEB	C	“Debug option”	D='ON'

Possible IDEB values are:

ON

OFF

If ideb='ON ' then :

```
idebug=1, idemin=1, idemax=1000000, itime=1
```

else :

```
idebug=0, idemin=0, idemax=0
```

8 GEANT/RZ

ZEBRA/RZ commands.

8.1 PQUEST [iq1 iq2]

`IQ1`	I	"Lower limit for IQ index"	`D=1`
`IQ2`	I	"Upper limit for IQ index"	`D=20`

Print the array IQUEST in /QUEST/.

8.2 FILE lun fname [chopt]

`LUN`	I	"Logical unit number"	
`FNAME`	C	"File name"	
`CHOPT`	C	"Options"	`D='␣'`

Possible `CHOPT` values are:

`'␣'`

`A`

`N`

`U`

Open a GRZ file.

```
CHOPT=' ' readonly mode
CHOPT='U' update mode
CHOPT='N' create new file
CHOPT='I' Read all structures from existing file
CHOPT='O' Write all structures on file
```

8.3 REND lunrz

`LUNRZ`	I	"Logical unit number"

Close an RZ file opened by GRFILE on logical unit LUNRZ.

```
CALL GREND(LUNRZ)
```

8.4 MDIR chdir [chopt]

`CHDIR`	C	"Directory name"	
`CHOPT`	C	"Options"	`D='␣'`

To create a new RZ directory below the current directory. with

```
RZTAGS(1)='Object'
RZTAGS(2)='Idvers-NR '
```

8.5 CDIR [chpath chopt]

```
CHPATH   C   “Path name”  D='␣'
CHOPT    C   “CHOPT”  D='␣'
```

Change or print the current directory.

```
Ex.  CD dir1         ; make DIR1 the new CWD
     CD //file1/dir2 ; make //FILE1/DIR2 the new CWD
     CD              ; print the name of the CWD
```

8.6 IN object [idvers chopt]

```
OBJECT   C   “Structure name”
IDVERS   I   “Version number”  D=1
CHOPT    C   “Option”  D='␣'
```

Read data structure identified by OBJECT,IDVERS into memory.

```
MATE read JMATE structure
TMED read JTMED structure
VOLU read JVOLUM structure
ROTM read JROTM structure
SETS read JSET  structure
PART read JPART structure
SCAN read LSCAN structure
INIT read all above data structures
```

8.7 OUT object [idvers chopt]

```
OBJECT   C   “Structure name”
IDVERS   I   “Version number”  D=1
CHOPT    C   “Option”  D='␣'
```

Write data structure identified by OBJECT,IDVERS to RZ file.

```
MATE write JMATE structure
TMED write JTMED structure
VOLU write JVOLUM structure
ROTM write JROTM structure
SETS write JSET  structure
PART write JPART structure
SCAN write LSCAN structure
INIT write all above data structures
```

8.8 LDIR [chpath chopt]

```
CHPATH   C   “Path name”  D='␣'
CHOPT    C   “CHOPT”  D='␣'
```

List the contents of a directory (memory or disk). To list all RZ files currently open, type 'LD //'.

8.9 PURGE [nkeep]

`NKEEP` I "Number of cycles to keep" `D=1`

Purge an RZ directory.

8.10 SCR object [idvers]

`OBJECT` C "Structure name"
`IDVERS` I "Version number" `D=1`

Delete entry identified by OBJECT,IDVERS on RZ file. OBJECT may be : MATE,TMED,VOLU,ROTM,SETS,PART,SCAN, * If OBJECT= * delete all entries with IDVERS.

8.11 LOCK chdir

`CHDIR` C "Lock identifier" `D='RZFILE'`

Lock an RZ directory.

8.12 FREE chdir

`CHDIR` C "Lock identifier" `D='RZFILE'`

Free an RZ directory.

9 GEANT/FZ

ZEBRA/FZ commands

9.1 FZIN lun keysu [ident]

`LUN`	I	"Fortran unit of the FZ file"
`KEYSU`	C	"Name of the data structure to be retrieved"
`IDENT`	I	"Version of the data structure to be retrieved" `D=0`

Equivalent to a call to:

```
CALL GFIN(LUN,KEYSU,1,IDENT,' ',IER)
```

9.2 FZOPEN lun file luntyp len [chopt]

`LUN`	I	"Fortran unit with which to open the file"
`FILE`	C	"Name of the file to be opened"
`LUNTYP`	C	"Type of FZ file to be opened by GOPEN" `D='XI'`
`LEN`	I	"Recordlenght of the file" `D=0`
`CHOPT`	C	"Optional parameter to specify the action" `D='␣'`

Equivalent to a call to:

```
CALL GOPEN(LUN,FILE,LUNTYP,LEN,IER)
```

If CHOPT = I then a call to GFIN or GFOUT will be performed in addition according to the value of LUNTYP, with the key INIT to save or retrieve the whole initialization data structure.

9.3 FZOUT lun keysu [ident]

`LUN`	I	"Fortran unit of the FZ file"
`KEYSU`	C	"Name of the data structure to be saved"
`IDENT`	I	"Version of the data structure to be saved" `D=1`

Equivalent to a call to:

```
CALL GFOUT(LUN,KEYSU,1,IDENT,' ',IER)
```

9.4 FZCLOSE lun

`LUN`	I	"Fortran unit of the FZ to close"

Equivalent to a call to:

```
CALL GCLOSE(LUN,IER)
```

10 GEANT/DZ

10.1 SURV name [number]

NAME	C	"Bank name"
NUMBER	I	"Bank number" D=1

Print a survey of the structure identified by NAME, NUMBER.

10.2 SHOW name [number chopt]

NAME	C	"Bank name"
NUMBER	I	"Bank number" D=1
CHOPT	C	"Options" D='BSV'

Display the contents of a bank or a data structure identified by its NAME and NUMBER. The output format of the data part is controlled by the internal or external I/O characteristic.

```
CHOPT='B' Print the bank.
CHOPT='S' Print the bank contents from left to right Sideways
          with up to ten elements per line.
CHOPT='V' Print the vertical (down) structure.
CHOPT='D' Print the bank contents from top to bottom Downwards
          with five elements per line.
CHOPT='L' Print the linear structure.
CHOPT='Z' Print the data part of each bank in hexadecimal format
```

10.3 SNAP [idiv chopt]

IDIV	I	"Division number " D=2 R=0:24
CHOPT	C	"Options" D='M'

Snap of one or more divisions. Provides a snapshot of one or more divisions in a ZEBRA store. The kind of information provided is controlled by CHOPT.

```
CHOPT='M' Print Map entry for each bank
CHOPT='E' Extend map entry to dump all links of each bank
          (otherwise only as many links as will fit on a line)
CHOPT='F' Full. Dump all active banks, links and data
CHOPT='K' Kill. Dropped banks to be treated as active
          (dropped banks are not normally dumped under D or F option)
CHOPT='L' Dump all Link areas associated with the store
CHOPT='W' Dump the Working space, links and data
CHOPT='Z' Dump the information in hexadecimal.
```

10.4 VERIFY [idiv chopt]

IDIV	I	"Division number " D=0 R=0:24
CHOPT	C	"Options" D='CLSU'

Check the structure of one or more ZEBRA divisions. The verification detail depends on the settings in CHOPT.

```
CHOPT='C' Check chaining of banks only
CHOPT='L' Check validity of the structural links (implies 'C')
CHOPT='S' Check the store parameters
CHOPT='U' Check the validity of the up and origin (implies 'C')
CHOPT='F' Errors are considered fatal and generate a call to ZFATAL
```

10.5 STORE [ixstor]

IXSTOR	I	"Store number"	D=0 R=0:24

Display the structure of the ZEBRA store IXSTOR. Output the parameters characterizing the store, followed by a list of all divisions and all link areas associated with the store in question.

10.6 DDIV [idiv path]

IDIV	I	"Division number"	D=2
PATH	C	"Name of the doc file"	D='␣'

Facility to display the layout of stores and divisions.

```
CALL DZDDIV(idiv,LDUMMY,path,'IN',1,0,1,IWTYPE)
```

10.7 DISP bank [path number]

BANK	C	"Name of the bank"	
PATH	C	"Name of the doc file"	D='␣'
NUMBER	I	"Number of the bank"	D=1

Interactive bank display tool.

```
CALL DZDISP(IXSTOR,LBANK,path,'N',1,0,1,IWTYPE)
```

10.8 DIRZ [path]

PATH	C	"Name of the RZ directory to analyse"

Facility to display RZ directory trees.

```
CALL DZDIRZ(0,LDUMMY,0,path,'N',1,0,1)
```

11 GEANT/SCAN

To define parameters for the SCAN geometry. If the routine GUSTEP and GUKINE are properly instrumented (see examples in GEANX), when the TRI command is entered NTETA Geantinos will be tracked through the real detector starting at the vertex position defined by the command vertex. A simplified version of the geometry is automatically generated in (ETA,PHI) or (THETA,PHI) following the option given in the command TETA. The data structure LSCAN generated may be saved on an RZ file for subsequent processing. This data structure may be used for fast parametrization techniques.

11.1 PHI nphi [phimin phimax]

`NPHI`	I	"Number of PHI divisions"	`D=90`
`PHIMIN`	R	"Minimum PHI in degrees"	`D=0.  R=0.:360.`
`PHIMAX`	R	"Maximum PHI in degrees"	`D=360.  R=0.:360.`

To specify number of divisions along PHI. If no parameter is given, the current values of the parameters are displayed.

11.2 TETA nteta [tetmin tetmax divtyp]

`NTETA`	I	"Number of TETA divisions"	`D=90`
`TETMIN`	R	"Minimum value of TETA"	
`TETMAX`	R	"Maximum value of TETA"	
`DIVTYP`	I	"Type of TETA division"	`R=1:3`

To specify number of divisions along TETA. If DIVTYP=1 divisions in pseudo-rapidity ETA. If DIVTYP=2 divisions in degrees following the THETA angle. If DIVTYP=3 divisions in cos(TETA). If no parameter is given, the current values of the parameters are displayed.

11.3 SLIST list

`LIST`	C	"List of master volumes"

Only boundary crossings of volumes given in LIST will be seen in the SCAN geometry. If no parameters are given, the current SCAN volumes will be listed. If a full stop (.) is given, the list of scan volumes will be erased.

11.4 VERTEX vx vy vz

`VX`	R	"Scan X-origin"	`D=0.`
`VY`	R	"Scan Y-origin"	`D=0.`
`VZ`	R	"Scan Z-origin"	`D=0.`

All Geantinos tracked will start from position VX,VY,VZ.

11.5 SFACTORS factx0 factl factr

`FACTX0`	R	"Scale factor for SX0"	`D=100.`

FACTL	R	"Scale factor for SL" D=1000.
FACTR	R	"Scale factor for R" D=100.

Set scale factors for SX0,SL and R. The given scale factors must be such that:

```
SX0*FACTX0 < 2**15-1 (32767)
SL*FACTL   < 2**10-1 (1023)
SR*FACTR   < 2**17-1 (131071)
```

11.6 STURN chopt

CHOPT	C	"SCAN mode setting"

Possible CHOPT values are:

ON

OFF

INIT

Switch on/off SCAN mode. If SCAN mode is on, SCAN geantinos are generated and tracked to fill (or complete) the current scan data structure. If SCAN mode is off, normal kinematics generation and tracking will take place. If INIT is given, the current SCAN data structure (if any) will be dropped and SCAN mode will be turned on.

11.7 PCUTS [iparam pcutga pcutel pcutha pcutne pcutmu]

IPARAM	I	"Parametrization Flag" R=0:1
PCUTGA	R	"Parametrization Cut for gammas"
PCUTEL	R	"Parametrization Cut for electrons"
PCUTHA	R	"Parametrization Cut for charged hadrons"
PCUTNE	R	"Parametrization Cut for neutral hadrons"
PCUTMU	R	"Parametrization Cut for muons"

Control parametrization at tracking time.

```
IPARAM=0       No parametrization is performed
IPARAM=1       Parametrization is performed
```

If parametrization is active and a particle falls below its parametrization cut, then the particle will be replaced by a parametrized shower which will be tracked in the SCAN geometry.

11.8 LSCAN id [volume chopt]

ID	I	"Lego plot identifier" D=2000
VOLUME	C	"Volume name" D='XXXX'
CHOPT	C	"List of options" D='OPX'

Possible CHOPT values are:

'␣'

O

P

I

X

L

Generates and plot a table of physics quantities such as the total number of radiation lengths or interaction lengths in function of the SCAN parameters TETA,PHI.

```
CHOPT='O' table is generated at Exit  of VOLUME.
CHOPT='I' table is generated at Entry of VOLUME.
CHOPT='X' radiation lengths
CHOPT='L' Interaction lengths
CHOPT='P' Plot the table
```

If VOLUME='XXXX' Mother volume is used.

11.9 HSCAN idphi [volume chopt]

IDPHI	I	"Histogram/phi identifier"	D=1000
VOLUME	C	"Volume name"	D='XXXX'
CHOPT	C	"List of options"	D='OPX'

Possible CHOPT values are:

'␣'

O

P

I

X

L

Generates and plots an histogram of physics quantities such as the total number of radiation lengths or interaction lengths as a function of the SCAN parameter TETA for a given value of PHI.

```
CHOPT='O' histogram is generated at Exit  of VOLUME.
CHOPT='I' histogram is generated at Entry of VOLUME.
CHOPT='X' radiation lengths
CHOPT='L' Interaction lengths
CHOPT='P' Plot the histogram
```

If VOLUME='XXXX' Mother volume is used. The histogram identifier IDPHI is used to also identify which PHI division to plot: IPHI=MOD(IDPHI,1000). If IPHI=0, then all PHI divisions are generated (not plotted) with histogram identifiers IDPHI+PHI division number.

12 GEANT/PHYSICS

Commands to set physics parameters.

12.1 ANNI [ianni]

`IANNI` C "Flag IANNI" `D='1'`

Possible `IANNI` values are:

`0`

`1`

`2`

To control positron annihilation.

```
IANNI=0 no annihilation
     =1 annihilation. Decays processed.
     =2 annihilation. No decay products stored.
```

12.2 AUTO [iauto]

`IAUTO` C "Flag IAUTO" `D='1'`

Possible `IAUTO` values are:

`0`

`1`

To control automatic calculation of tracking medium parameters:

```
IAUTO=0 no automatic calculation;
     =1 automati calculation.
```

12.3 BREM [ibrem]

`IBREM` C "Flag IBREM" `D='1'`

Possible `IBREM` values are:

`0`

`1`

`2`

To control bremstrahlung.

```
IBREM=0 no bremstrahlung
     =1 bremstrahlung. Photon processed.
     =2 bremstrahlung. No photon stored.
```

12.4 CKOV [ickov]

`ICKOV` C "Flag ICKOV" `D='0'`

Possible `ICKOV` values are:

0

1

2

To control Cerenkov production

```
ICOMP=0 no Cerenkov;
     =1 Cerenkov;
     =2 Cerenkov with primary stopped at each step.
```

12.5 COMP [icomp]

`ICOMP` C “Flag ICOMP” `D='1'`

Possible `ICOMP` values are:

0

1

2

To control Compton scattering

```
ICOMP=0 no Compton
     =1 Compton. Electron processed.
     =2 Compton. No electron stored.
```

12.6 DCAY [idcay]

`IDCAY` C “Flag IDCAY” `D='1'`

Possible `IDCAY` values are:

0

1

2

To control Decay mechanism.

```
IDCAY=0 no decays.
     =1 Decays. secondaries processed.
     =2 Decays. No secondaries stored.
```

12.7 DRAY [idray]

`IDRAY` C “Flag IDRAY” `D='1'`

Possible `IDRAY` values are:

0

1

2

To control delta rays mechanism.

```
IDRAY=0 no delta rays.
     =1 Delta rays. secondaries processed.
     =2 Delta rays. No secondaries stored.
```

12.8 ERAN [ekmin ekmax nekbin]

`EKMIN`	R	"Minimum energy of the tables" `D=1E-5`
`EKMAX`	R	"Maximum energy of the tables" `D=1E+4`
`NEKBIN`	I	"Number of bins in the tables" `D=90 R=1:200`

To define the range and binning of internal tables.

12.9 HADR [ihadr]

`IHADR`	I	"Flag IHADR" `D=1`

To control hadronic interactions.

```
IHADR=0 no hadronic interactions.
     =1 Hadronic interactions. secondaries processed.
     =2 Hadronic interactions. No secondaries stored.
```

12.10 LABS [labs]

`LABS`	I	"Flag LABS" `D=0`

To control absorbtion of Cerenkov photons:

```
LABS=0 no absorbtion of photons;
LABS=1 absorbtion of photons;
```

12.11 LOSS [iloss]

`ILOSS`	C	"Flag ILOSS" `D='2'`

Possible `ILOSS` values are:

0

1

2

3

4

To control energy loss.

```
ILOSS=0 no energy loss;
     =1 restricted energy loss fluctuations;
     =2 complete energy loss fluctuations;
     =3 same as 1;
     =4 no energy loss fluctuations.
```

If the value ILOSS is changed, then cross-sections and energy loss tables must be recomputed via the command 'PHYSI'.

12.12 MULS [imuls]

`IMULS` C "Flag IMULS" `D='1'`

Possible `IMULS` values are:

`0`

`1`

`2`

`3`

To control multiple scattering.

```
IMULS=0 no multiple scattering.
     =1 Moliere or Coulomb scattering.
     =2 Moliere or Coulomb scattering.
     =3 Gaussian scattering.
```

12.13 MUNU [imunu]

`IMUNU` C "Flag IMUNU" `D='1'`

Possible `IMUNU` values are:

`0`

`1`

`2`

To control muon nuclear interactions.

```
IMUNU=0 no muon-nuclear interactions.
     =1 Nuclear interactions. Secondaries processed.
     =2 Nuclear interactions. Secondaries not processed.
```

12.14 PAIR [ipair]

`IPAIR` C "Flag IPAIR" `D='1'`

Possible `IPAIR` values are:

0

1

2

To control pair production mechanism.

```
IPAIR=0 no pair production.
     =1 Pair production. secondaries processed.
     =2 Pair production. No secondaries stored.
```

12.15 PFIS [ipfis]

`IPFIS` C "Flag IPFIS" `D='1'`

Possible `IPFIS` values are:

0

1

2

To control photo fission mechanism.

```
IPFIS=0 no photo fission.
     =1 Photo fission. secondaries processed.
     =2 Photo fission. No secondaries stored.
```

12.16 PHOT [iphot]

`IPHOT` C "Flag IPHOT" `D='1'`

Possible `IPHOT` values are:

0

1

2

To control Photo effect.

```
IPHOT=0 no photo electric effect.
     =1 Photo effect. Electron processed.
     =2 Photo effect. No electron stored.
```

12.17 RAYL [irayl]

`IRAYL` C "Flag IRAYL" `D='1'`

Possible `IRAYL` values are:

0

1

To control Rayleigh scattering.

```
IRAYL=0 no Rayleigh scattering.
     =1 Rayleigh.
```

12.18 STRA [istra]

ISTRA	C	"Flag ISTRA"	D='0'

Possible ISTRA values are:

0

1

2

To control energy loss fluctuation model:

```
ISTRA=0 Urban model;
     =1 PAI model;
     =2 PAI+ASHO model (not active at the moment).
```

12.19 SYNC [isync]

ISYNC	C	"Flag ISYNC"	D='1'

Possible ISYNC values are:

0

1

To control synchrotron radiation:

```
ISYNC=0 no synchrotron radiation;
     =1 synchrotron radiation.
```

12.20 CUTS [cutgam cutele cuthad cutneu cutmuo bcute bcutm dcute dcutm ppcutm tofmax gcuts]

CUTGAM	R	"Cut for gammas"	D=0.001
CUTELE	R	"Cut for electrons"	D=0.001
CUTHAD	R	"Cut for charged hadrons"	D=0.01
CUTNEU	R	"Cut for neutral hadrons"	D=0.01
CUTMUO	R	"Cut for muons"	D=0.01
BCUTE	R	"Cut for electron brems."	D=-1.
BCUTM	R	"Cut for muon brems."	D=-1.
DCUTE	R	"Cut for electron delta-rays"	D=-1.
DCUTM	R	"Cut for muon delta-rays"	D=-1.
PPCUTM	R	"Cut for e+e- pairs by muons"	D=0.01
TOFMAX	R	"Time of flight cut"	D=1.E+10
GCUTS	R	"5 user words"	D=0.

To change physics cuts. If no parameter is given, the list of the current cuts is printed.

```
If the default values (-1.) for       BCUTE ,BCUTM ,DCUTE ,DCUTM
are not modified, they will be set to CUTGAM,CUTGAM,CUTELE,CUTELE
respectively.
```

If one of the parameters from CUTGAM to PPCUTM included is modified, cross-sections and energy loss tables must be recomputed via the command 'PHYSI'.

12.21 DRPRT ipart imate step [npoint]

IPART	I	"GEANT particle number"
IMATE	I	"GEANT material number"
STEP	R	"step length in centimeters"
NPOINT	I	"number of logarithmically spaced energy points" D=10 R=2:100

This routine prints the relevant parameters linked with the energy loss fluctuation.

12.22 PHYSI

Call the GEANT initialisation routine GPHYSI to recompute the tables of cross-sections and energy loss. This command must be invoked after CUTS, LOSS or ERAN commands.

13 GEANT/LISTS

13.1 HSTA [lhsta_1 lhsta_2 lhsta_3 lhsta_4 lhsta_5 lhsta_6 lhsta_7 lhsta_8 lhsta_9 lhsta_10 lhsta_11 lhsta_12 lhsta_13 lhsta_14 lhsta_15 lhsta_16 lhsta_17 lhsta_18 lhsta_19 lhsta_20]

LHSTA_1	C	"user word"
LHSTA_2	C	"user word"
LHSTA_3	C	"user word"
LHSTA_4	C	"user word"
LHSTA_5	C	"user word"
LHSTA_6	C	"user word"
LHSTA_7	C	"user word"
LHSTA_8	C	"user word"
LHSTA_9	C	"user word"
LHSTA_10	C	"user word"
LHSTA_11	C	"user word"
LHSTA_12	C	"user word"
LHSTA_13	C	"user word"
LHSTA_14	C	"user word"
LHSTA_15	C	"user word"
LHSTA_16	C	"user word"
LHSTA_17	C	"user word"
LHSTA_18	C	"user word"
LHSTA_19	C	"user word"
LHSTA_20	C	"user word"

The command HSTA is similar to the HSTA data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.2 GET [lget_1 lget_2 lget_3 lget_4 lget_5 lget_6 lget_7 lget_8 lget_9 lget_10 lget_11 lget_12 lget_13 lget_14 lget_15 lget_16 lget_17 lget_18 lget_19 lget_20]

LGET_1	C	"user word"
LGET_2	C	"user word"
LGET_3	C	"user word"
LGET_4	C	"user word"
LGET_5	C	"user word"
LGET_6	C	"user word"
LGET_7	C	"user word"
LGET_8	C	"user word"
LGET_9	C	"user word"
LGET_10	C	"user word"
LGET_11	C	"user word"
LGET_12	C	"user word"
LGET_13	C	"user word"
LGET_14	C	"user word"

LGET_15	C	"user word"
LGET_16	C	"user word"
LGET_17	C	"user word"
LGET_18	C	"user word"
LGET_19	C	"user word"
LGET_20	C	"user word"

The command GET is similar to the GET data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.3 SAVE [lsave_1 lsave_2 lsave_3 lsave_4 lsave_5 lsave_6 lsave_7 lsave_8 lsave_9 lsave_10 lsave_11 lsave_12 lsave_13 lsave_14 lsave_15 lsave_16 lsave_17 lsave_18 lsave_19 lsave_20]

LSAVE_1	C	"user word"
LSAVE_2	C	"user word"
LSAVE_3	C	"user word"
LSAVE_4	C	"user word"
LSAVE_5	C	"user word"
LSAVE_6	C	"user word"
LSAVE_7	C	"user word"
LSAVE_8	C	"user word"
LSAVE_9	C	"user word"
LSAVE_10	C	"user word"
LSAVE_11	C	"user word"
LSAVE_12	C	"user word"
LSAVE_13	C	"user word"
LSAVE_14	C	"user word"
LSAVE_15	C	"user word"
LSAVE_16	C	"user word"
LSAVE_17	C	"user word"
LSAVE_18	C	"user word"
LSAVE_19	C	"user word"
LSAVE_20	C	"user word"

The command SAVE is similar to the SAVE data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.4 SETS [lsets_1 lsets_2 lsets_3 lsets_4 lsets_5 lsets_6 lsets_7 lsets_8 lsets_9 lsets_10 lsets_11 lsets_12 lsets_13 lsets_14 lsets_15 lsets_16 lsets_17 lsets_18 lsets_19 lsets_20]

LSETS_1	C	"user word"
LSETS_2	C	"user word"
LSETS_3	C	"user word"
LSETS_4	C	"user word"
LSETS_5	C	"user word"

LSETS_6	C	"user word"
LSETS_7	C	"user word"
LSETS_8	C	"user word"
LSETS_9	C	"user word"
LSETS_10	C	"user word"
LSETS_11	C	"user word"
LSETS_12	C	"user word"
LSETS_13	C	"user word"
LSETS_14	C	"user word"
LSETS_15	C	"user word"
LSETS_16	C	"user word"
LSETS_17	C	"user word"
LSETS_18	C	"user word"
LSETS_19	C	"user word"
LSETS_20	C	"user word"

The command SETS is similar to the SETS data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.5 LPRIN [lprin_1 lprin_2 lprin_3 lprin_4 lprin_5 lprin_6 lprin_7 lprin_8 lprin_9 lprin_10 lprin_11 lprin_12 lprin_13 lprin_14 lprin_15 lprin_16 lprin_17 lprin_18 lprin_19 lprin_20]

LPRIN_1	C	"user word"
LPRIN_2	C	"user word"
LPRIN_3	C	"user word"
LPRIN_4	C	"user word"
LPRIN_5	C	"user word"
LPRIN_6	C	"user word"
LPRIN_7	C	"user word"
LPRIN_8	C	"user word"
LPRIN_9	C	"user word"
LPRIN_10	C	"user word"
LPRIN_11	C	"user word"
LPRIN_12	C	"user word"
LPRIN_13	C	"user word"
LPRIN_14	C	"user word"
LPRIN_15	C	"user word"
LPRIN_16	C	"user word"
LPRIN_17	C	"user word"
LPRIN_18	C	"user word"
LPRIN_19	C	"user word"
LPRIN_20	C	"user word"

The command PRIN is similar to the PRIN data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.6 GEOM [lgeom_1 lgeom_2 lgeom_3 lgeom_4 lgeom_5 lgeom_6 lgeom_7 lgeom_8 lgeom_9 lgeom_10 lgeom_11 lgeom_12 lgeom_13 lgeom_14 lgeom_15 lgeom_16 lgeom_17 lgeom_18 lgeom_19 lgeom_20]

`LGEOM_1`	C	"user word"
`LGEOM_2`	C	"user word"
`LGEOM_3`	C	"user word"
`LGEOM_4`	C	"user word"
`LGEOM_5`	C	"user word"
`LGEOM_6`	C	"user word"
`LGEOM_7`	C	"user word"
`LGEOM_8`	C	"user word"
`LGEOM_9`	C	"user word"
`LGEOM_10`	C	"user word"
`LGEOM_11`	C	"user word"
`LGEOM_12`	C	"user word"
`LGEOM_13`	C	"user word"
`LGEOM_14`	C	"user word"
`LGEOM_15`	C	"user word"
`LGEOM_16`	C	"user word"
`LGEOM_17`	C	"user word"
`LGEOM_18`	C	"user word"
`LGEOM_19`	C	"user word"
`LGEOM_20`	C	"user word"

The command GEOM is similar to the GEOM data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.7 VIEW [lview_1 lview_2 lview_3 lview_4 lview_5 lview_6 lview_7 lview_8 lview_9 lview_10 lview_11 lview_12 lview_13 lview_14 lview_15 lview_16 lview_17 lview_18 lview_19 lview_20]

`LVIEW_1`	C	"user word"
`LVIEW_2`	C	"user word"
`LVIEW_3`	C	"user word"
`LVIEW_4`	C	"user word"
`LVIEW_5`	C	"user word"
`LVIEW_6`	C	"user word"
`LVIEW_7`	C	"user word"
`LVIEW_8`	C	"user word"
`LVIEW_9`	C	"user word"
`LVIEW_10`	C	"user word"
`LVIEW_11`	C	"user word"
`LVIEW_12`	C	"user word"
`LVIEW_13`	C	"user word"
`LVIEW_14`	C	"user word"

LVIEW_15	C	"user word"
LVIEW_16	C	"user word"
LVIEW_17	C	"user word"
LVIEW_18	C	"user word"
LVIEW_19	C	"user word"
LVIEW_20	C	"user word"

The command VIEW is similar to the VIEW data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.8 PLOT [lplot_1 lplot_2 lplot_3 lplot_4 lplot_5 lplot_6 lplot_7 lplot_8 lplot_9 lplot_10 lplot_11 lplot_12 lplot_13 lplot_14 lplot_15 lplot_16 lplot_17 lplot_18 lplot_19 lplot_20]

LPLOT_1	C	"user word"
LPLOT_2	C	"user word"
LPLOT_3	C	"user word"
LPLOT_4	C	"user word"
LPLOT_5	C	"user word"
LPLOT_6	C	"user word"
LPLOT_7	C	"user word"
LPLOT_8	C	"user word"
LPLOT_9	C	"user word"
LPLOT_10	C	"user word"
LPLOT_11	C	"user word"
LPLOT_12	C	"user word"
LPLOT_13	C	"user word"
LPLOT_14	C	"user word"
LPLOT_15	C	"user word"
LPLOT_16	C	"user word"
LPLOT_17	C	"user word"
LPLOT_18	C	"user word"
LPLOT_19	C	"user word"
LPLOT_20	C	"user word"

The command PLOT is similar to the PLOT data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.9 STAT [lstat_1 lstat_2 lstat_3 lstat_4 lstat_5 lstat_6 lstat_7 lstat_8 lstat_9 lstat_10 lstat_11 lstat_12 lstat_13 lstat_14 lstat_15 lstat_16 lstat_17 lstat_18 lstat_19 lstat_20]

LSTAT_1	C	"user word"
LSTAT_2	C	"user word"
LSTAT_3	C	"user word"
LSTAT_4	C	"user word"
LSTAT_5	C	"user word"
LSTAT_6	C	"user word"

`LSTAT_7`	C	"user word"
`LSTAT_8`	C	"user word"
`LSTAT_9`	C	"user word"
`LSTAT_10`	C	"user word"
`LSTAT_11`	C	"user word"
`LSTAT_12`	C	"user word"
`LSTAT_13`	C	"user word"
`LSTAT_14`	C	"user word"
`LSTAT_15`	C	"user word"
`LSTAT_16`	C	"user word"
`LSTAT_17`	C	"user word"
`LSTAT_18`	C	"user word"
`LSTAT_19`	C	"user word"
`LSTAT_20`	C	"user word"

The command STAT is similar to the STAT data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.10 RGET [lrget_1 lrget_2 lrget_3 lrget_4 lrget_5 lrget_6 lrget_7 lrget_8 lrget_9 lrget_10 lrget_11 lrget_12 lrget_13 lrget_14 lrget_15 lrget_16 lrget_17 lrget_18 lrget_19 lrget_20]

`LRGET_1`	C	"user word"
`LRGET_2`	C	"user word"
`LRGET_3`	C	"user word"
`LRGET_4`	C	"user word"
`LRGET_5`	C	"user word"
`LRGET_6`	C	"user word"
`LRGET_7`	C	"user word"
`LRGET_8`	C	"user word"
`LRGET_9`	C	"user word"
`LRGET_10`	C	"user word"
`LRGET_11`	C	"user word"
`LRGET_12`	C	"user word"
`LRGET_13`	C	"user word"
`LRGET_14`	C	"user word"
`LRGET_15`	C	"user word"
`LRGET_16`	C	"user word"
`LRGET_17`	C	"user word"
`LRGET_18`	C	"user word"
`LRGET_19`	C	"user word"
`LRGET_20`	C	"user word"

The command RGET is similar to the RGET data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

13.11 RSAV [lrsave_1 lrsave_2 lrsave_3 lrsave_4 lrsave_5 lrsave_6 lrsave_7 lrsave_8 lrsave_9 lrsave_10 lrsave_11 lrsave_12 lrsave_13 lrsave_14 lrsave_15 lrsave_16 lrsave_17 lrsave_18 lrsave_19 lrsave_20]

`LRSAVE_1`	C	"user word"
`LRSAVE_2`	C	"user word"
`LRSAVE_3`	C	"user word"
`LRSAVE_4`	C	"user word"
`LRSAVE_5`	C	"user word"
`LRSAVE_6`	C	"user word"
`LRSAVE_7`	C	"user word"
`LRSAVE_8`	C	"user word"
`LRSAVE_9`	C	"user word"
`LRSAVE_10`	C	"user word"
`LRSAVE_11`	C	"user word"
`LRSAVE_12`	C	"user word"
`LRSAVE_13`	C	"user word"
`LRSAVE_14`	C	"user word"
`LRSAVE_15`	C	"user word"
`LRSAVE_16`	C	"user word"
`LRSAVE_17`	C	"user word"
`LRSAVE_18`	C	"user word"
`LRSAVE_19`	C	"user word"
`LRSAVE_20`	C	"user word"

The command RSAV is similar to the RSAV data records. It can accept up to 20 4-character words. If the first argument is '.', the number of words is reset to 0 and all the words to four blanks.

Origin :
Revision :
Documentation : S.Ravndal

Submitted: 17.03.94
Revised: 17.03.94

Screen views of GEANT++

In this section some examples of the GEANT++ Motif interface are presented.

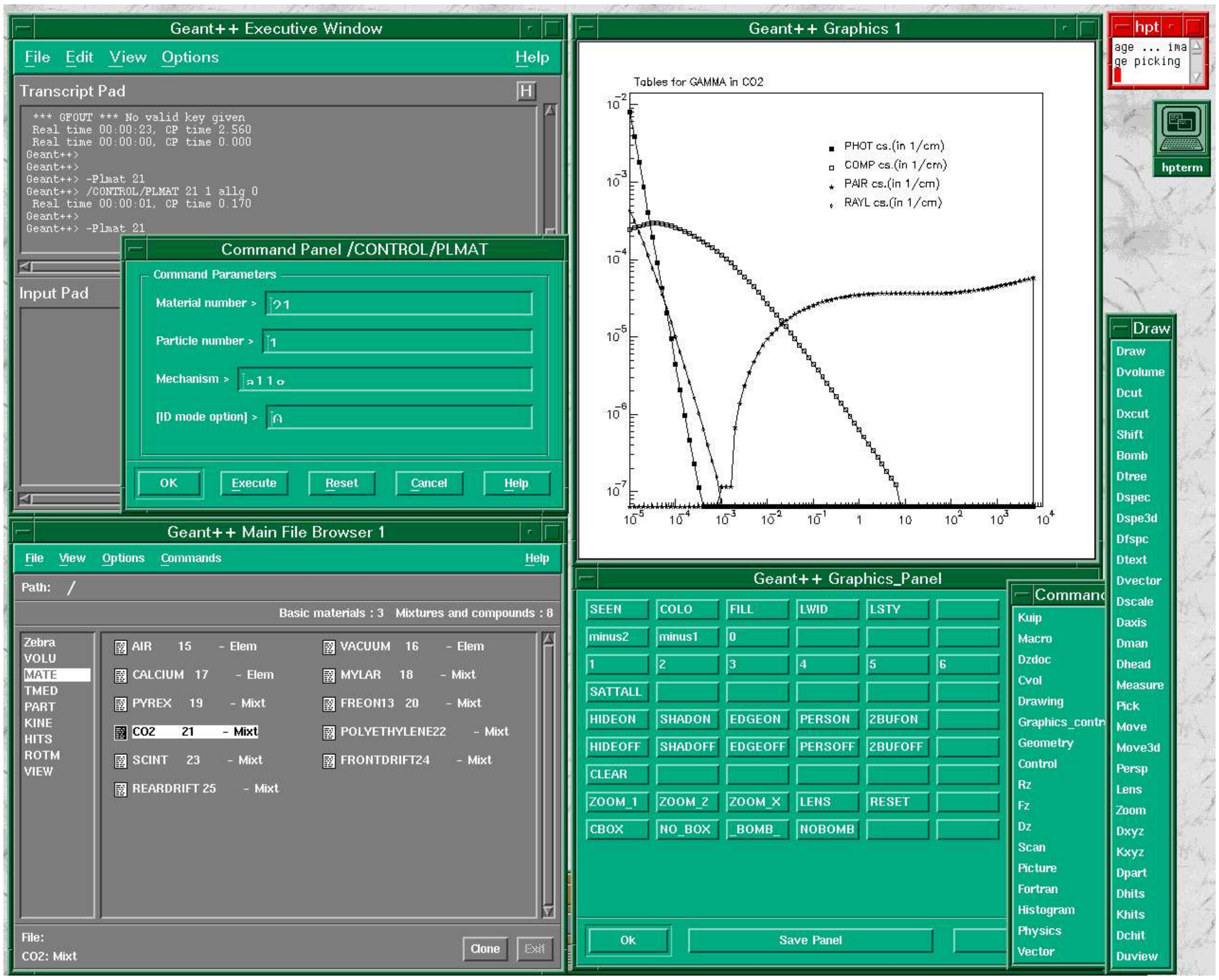


Figure 60: Plotting cross-sections of a material

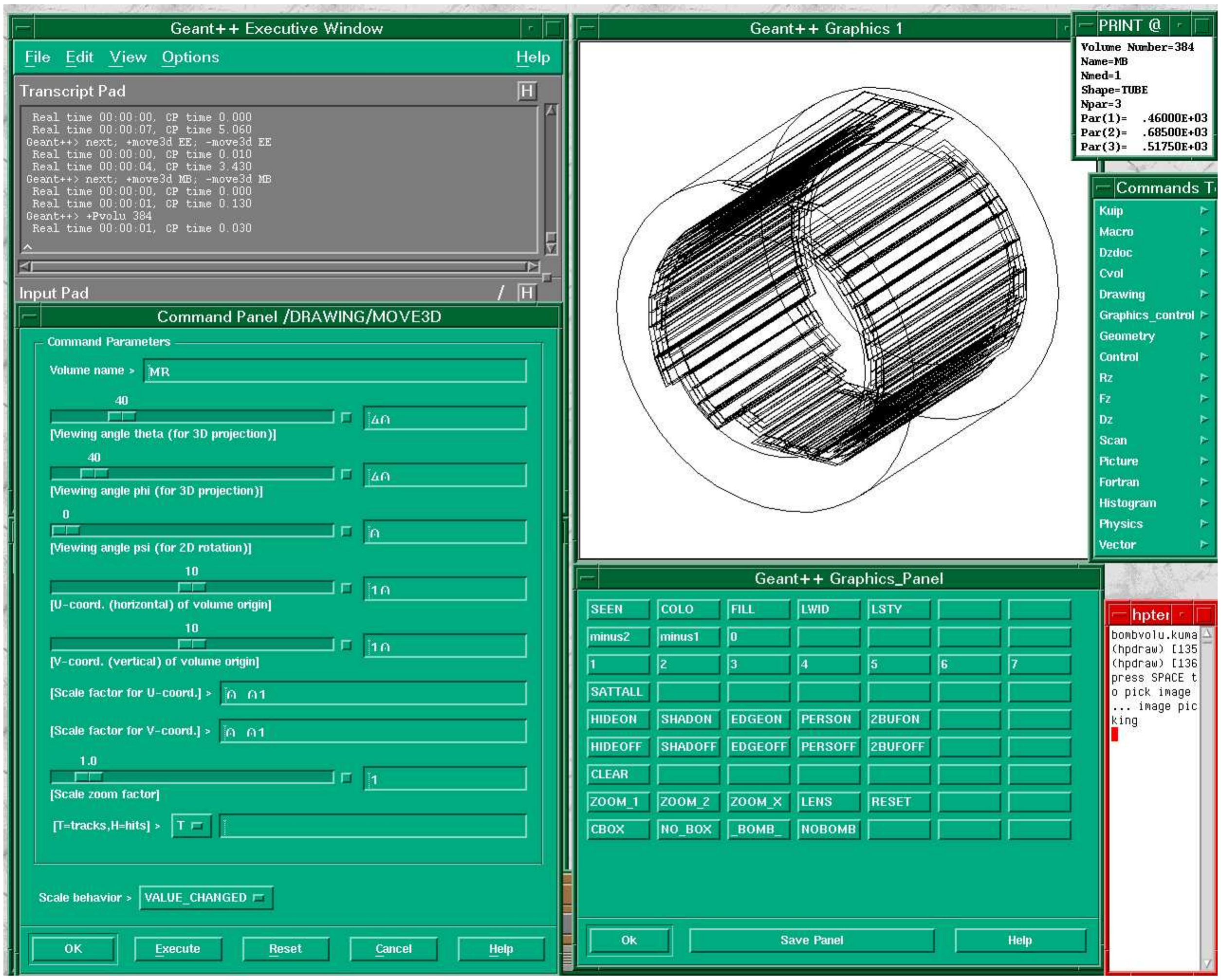


Figure 61: Draw using the MOVE 3D panel

XINT Bibliography

[1] R.Brun and P.Zanarini. *KUIP – Kit for a User Interface Package,* Program library I202. CERN, 1988.

Origin : GEANT **Submitted:** 01.10.84
Revision : **Revised:** 09.09.87
Documentation :

Index of Documented GEANT routines

```
GDPART DRAW001, DRAW130, DRAW300
GDRAW  DRAW001, DRAW110, DRAW115, DRAW120
GDRAWC DRAW001, DRAW120, DRAW130, DRAW140, DRAW300, DRAW400
GDRAWT DRAW001, DRAW300, DRAW399, DRAW400
GDRAWV DRAW001, DRAW400
GDRAWX DRAW001, DRAW120, DRAW400
GDRAY  PHYS331, TRAK200
GDRPRT PHYS333
GDRELA PHYS100, PHYS330, PHYS430
GDRELE PHYS100, PHYS330
GDRELP PHYS100, PHYS430
GDRSGA PHYS100, PHYS430
GDRVOL DRAW001, DRAW115
GDSCAL DRAW001, DRAW300, DRAW400
GDSHOW DRAW001, DRAW300
GDSPEC DRAW001, DRAW220
GDTOM  GEOM320
GDTREE DRAW001, DRAW210
GDXYZ  DRAW001, DRAW115, DRAW130, DRAW300, DRAW400, TRAK499
GDZOOM DRAW001, DRAW110, DRAW115, DRAW120, DRAW400
GD3D3D DRAW001, DRAW400
GEDITV DRAW001
GFATT  GEOM500
GFDET  HITS100
GFDETA HITS105
GFDETD HITS120
GFDETH HITS110
GFDETU HITS130
GFDIGI HITS300, HITS399
GFFGO  BASE001, BASE040, BASE110, PHYS100
GFHITS HITS001, HITS200
GFINDS HITS200, TRAK001
GFKINE KINE001, KINE100, KINE199
GFMATE CONS001, CONS100
GFPARA GEOM500
GFPART CONS001, CONS300
GFPATH HITS001, HITS200
GFSTAT GEOM700
GFTMAT CONS001, CONS101
GFTMED CONS001, CONS200
GGCLOS GEOM900, HITS001
GGDETV GEOM900, HITS001
GGET   IOPA300, KINE001
GHEISH PHYS510
GHELIX CONS200, TRAK001, TRAK500
GHELX3 CONS200, TRAK001, TRAK500, XINT001, XINT110, XINT120,
       XINT130, XINT140, ZZZZ001, ZZZZ002, ZZZZ999
GHESIG PHYS510, TRAK001
GHMIX  CONS110
```

```
GHSTOP PHYS510
GICYL  HITS400
GINIT  BASE040, BASE110, CONS001, PHYS100
GINVOL GEOM001, GEOM300, GEOM700, TRAK001, TRAK130, TRAK200
GIPLAN HITS400
GLANDG PHYS332
GLANDO PHYS332
GLANDR PHYS332
GLAST  BASE001, BASE300, GEOM700
GLOREN PHYS410
GLOOK  BASE410
GLUDKY KINE210
GLUND  KINE200
GLUNDI BASE040, KINE200
GLVOLU GEOM001, GEOM300, GEOM600, HITS001
GMATE  BASE001, CONS001, CONS100
GMEDIA GEOM001, GEOM300, GEOM310, GEOM410, GEOM700, TRAK001
GMENEW GEOM001, GEOM300, GEOM410, TRAK130
GMOL   PHYS325, TRAK200
GMOLI  PHYS325
GMOLS  PHYS325
GMOL4  PHYS325
GMTOD  GEOM320
GMUNU  PHYS460, TRAK200
GMUNUI PHYS100, PHYS460
GMUSIG PHYS460
GNEXT  GEOM001, GEOM310, GEOM700, TRAK001, TRAK130
GNEXTS GEOM310
GNSOUT GEOM310
GOPEN  IOPA200
GOSEAR GEOM001, GEOM300, GEOM410
GPAIR  TRAK200
GPAIRG PHYS211
GPAIRM PHYS451, TRAK200
GPART  BASE001, CONS001, CONS300
GPCXYZ TRAK400
GPDIGI HITS300
GPDRIF HITS500
GPFIS  PHYS240, TRAK200
GPFISI PHYS100, PHYS240
GPGHEI PHYS510
GPHADR PHYS500
GPHITS HITS200
GPHOT  PHYS231, PHYS240, TRAK200
GPHOTI PHYS100, PHYS230
GPHSIG PHYS230
GPHYSI BASE001, BASE110, CONS199, PHYS100
GPJXYZ TRAK001, TRAK400
GPKINE KINE001, KINE100
```

```
GPLMAT CONS001, CONS102
GPMATE CONS001, CONS100
GPMWPC HITS500
GPPART CONS001, CONS300
GPRELA PHYS100, PHYS450
GPRELM PHYS100, PHYS450
GPRINT BASE400, CONS001
GPRMAT CONS001, CONS103
GPROBI CONS199
GPROTM GEOM200
GPRSGA PHYS100, PHYS210, PHYS450
GPRSGG PHYS100, PHYS210
GPRSGM PHYS100, PHYS450
GPSETS HITS100
GPSTAT GEOM700
GPTMED CONS001, CONS200
GPVERT KINE001, KINE100
GPVOLU GEOM100
GRDSGA PHYS330
GREND  IOPA400
GRFILE IOPA400
GRKUTA CONS200, TRAK001, TRAK500
GRUN   BASE001, BASE200, BASE299
GSAHIT DRAW140, HITS001, HITS200, HITS299
GSATT  DRAW110, DRAW115, DRAW120, DRAW130, DRAW210, DRAW220,
       DRAW500, DRAW510, GEOM500
GRGET  IOPA500
GRSAVE IOPA500
GSAVE  IOPA300
GSCHIT HITS001, HITS200, HITS299
GSCMED GEOM600
GSDET  HITS001, HITS100, HITS105, HITS110, HITS199, HITS200, HITS300, TRAK
GSDETA HITS001, HITS105, HITS199
GSDETD HITS001, HITS105, HITS120, HITS199, HITS300
GSDETH HITS001, HITS100, HITS105, HITS110, HITS120, HITS199, HITS200
GSDETU HITS001, HITS105, HITS130, HITS199
GSDETV GEOM900, HITS001, HITS100, HITS199
GSDIGI HITS001, HITS300, HITS399
GSDK   CONS001, CONS300, CONS310, CONS399, DRAW001, DRAW110,
       DRAW115, DRAW120, DRAW130, DRAW140, DRAW210, DRAW220,
       DRAW300, DRAW399, DRAW400, DRAW500, DRAW510, GEOM001, GEOM050
GSDVN  DRAW210, GEOM001, GEOM100, GEOM130, GEOM140
GSDVN2 GEOM001, GEOM150
GSDVT  DRAW210, GEOM001, GEOM100, GEOM140, GEOM199
GSDVT2 GEOM001, GEOM150
GSDVX  GEOM001, GEOM100, GEOM150, GEOM199
GSEAR  GEOM300, GEOM410
GSKINE KINE001, KINE100, KINE199
GSKING TRAK300, TRAK399
```

```
GSMATE CONS001, CONS100
GSMIXT CONS001, CONS110, CONS199
GSNEXT GEOM001, GEOM199, GEOM300, GEOM410
GSORD  GEOM001, GEOM199, GEOM300, GEOM400, GEOM900
GSPART CONS001, CONS300, CONS310, KINE100
GSPOS  DRAW210, GEOM001, GEOM110, GEOM130, GEOM199
GSPOSP DRAW210, GEOM001, GEOM100, GEOM120
GSROTM GEOM110, GEOM120, GEOM200, GEOM299
GSSEAR GEOM001, GEOM199, GEOM300, GEOM410
GSSTAK TRAK300
GSTMED CONS001, CONS200, TRAK001
GSTPAR CONS001, CONS210, CONS299, PHYS100
GSUSEA GEOM410
GSVERT KINE001, KINE100, KINE199, KINE200
GSVOLU GEOM001, GEOM100, GEOM110, GEOM120
GSXYZ  TRAK001, TRAK400, TRAK499
GTAU   KINE210, PHYS001, PHYS010
GTELEC CONS300, PHYS010, TRAK001, TRAK130, TRAK200
GTGAMA CONS300, PHYS010, TRAK001, TRAK130, TRAK200
GTHADR CONS300, PHYS010, TRAK001, TRAK130, TRAK200
GTMUON CONS300, PHYS010, TRAK001, TRAK130, TRAK200
GTNEUT CONS300, PHYS010, TRAK001, TRAK130, TRAK200
GTNINO CONS300, TRAK001, TRAK130, TRAK200
GTRACK TRAK001, TRAK120
GTREVE BASE001, DRAW140, TRAK001, TRAK110
GTRIG  BASE001, BASE200
GTRIGC BASE001, BASE200
GTRIGI BASE001, BASE200, BASE400
GTSET  PHYS010
GTVOL  TRAK001, TRAK130, TRAK200
GUDCAY CONS310, KINE210, PHYS400
GUDIGI BASE001, HITS001
GUFLD  BASE001, TRAK001, TRAK500
GUHADR BASE001, TRAK200
GUKINE BASE001, KINE200, KINE210
GUOUT  BASE001
GUPHAD BASE001, TRAK001, TRAK200
GUSEAR GEOM410
GUSTEP BASE001, DRAW140, HITS001, HITS100, TRAK001, TRAK130,
       TRAK300, TRAK400, TRAK499
GUSWIM BASE001, CONS200, TRAK001, TRAK200, TRAK500
GUTRAK TRAK001, TRAK110
GUTREV BASE001, TRAK001, TRAK110
GZEBRA BASE110
GZINIT BASE001, BASE110
```

ZZZZ Bibliography

Index