JOUKO VUOSKOSKI

NEUTRAL FILE FORMATS AND GEOMETRICAL INFORMATION

Preface

I would like to thank the director of the Research Institute for High Energy Physics (SEFT), Professor Risto Orava, for offering me the opportunity to work with funding from the Research Institute. I would like to thank my supervisor Docent Matti Lähdeniemi for his advice concerning this thesis and further studies. I would also like to thank Wolfram Roth, the project leader in SEFT, for his supervision.

The work leading to this thesis was carried out at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland. The work in the Application Software group of the Computing and Networks Division at CERN has been very fruitful. I would like to thank my group leader Dr. René Brun for making the environment as easy-going as possible. The help of René Brun has also reached my private life. I would like to thank my section leader and supervisor Dr. Federico Carminati for his advice and fine suggestions as well as his patience in working with me. His help has been valuable concerning my work in the GEANT-team and further studies.

I would like to thank Mik Ferran for his help and advice. The work in CADD-team has been very fruitful and rewarding and it has given me the knowledge needed for this thesis. The collaboration with Nils-Joar Hoimyr has been warm and cheerful with extension to our free time. I am also grateful to Simone Giani, Janusz Chwastowski, Ian Mclaren, Marthe Brun, Ulf Hordvik, Mikko Voipio and many, many others. Special thanks to Jonna Vuoskoski for giving me the strength needed to produce this thesis.

Geneva, 29.11.1991

Jouko Vuoskoski

Contents

Preface i

Contents ii

Abstract v

Appendices vi

Glossary ix

1 Introduction 1
- 1.1 Background 1
- 1.2 CERN 2
  - 1.2.1 Organization 3
- 1.3 Detectors 6
- 1.4 Purpose of the work 8
  - 1.4.1 The goal 9
  - 1.4.2 History 11
  - 1.4.3 Problems 12

2 GEANT — detector description and simulation package 15
- 2.1 General 15

2.2 The geometrical modeller of GEANT 17
2.2.1 The volume definition 17
2.2.2 Basic shapes 18
2.2.3 Creation of shapes 20
2.2.4 Data structure 21

3 3-D geometric modelling in CAD 24
3.1 Wire-frame model 25
3.2 Surface model 26
3.3 Solid models 29
3.3.1 Constructive solid geometry (CSG) 30
3.3.2 Boundary representation (B-rep) 31

4 SET —The neutral file format 33
4.1 General 33
4.2 SET -file format 36
4.3 Entity organization 38
4.4 The geometrical modelling via SET 39

5 SET - GEANT interface 42
5.1 General 42
5.2 Environment 43
5.3 Problems 44
5.3.1 The computing Environment 44
5.3.2 GEANT 45
5.3.3 SET 46
5.3.4 CAD 46
5.4 Solution 47
5.4.1 The principle of the interface 47
5.4.2 Writing the volumes 49
5.4.3 Handling of the positioning and rotation . . . 50
5.4.4 Tree decoding 51
5.4.5 Handling of the divisions 52

6 Operating **55**
6.1 General 55
6.2 Testing 55
6.2.1 Shapes 57
6.2.2 Positioning 57
6.2.3 Divisions 60
6.3 Exceptions 60
6.4 Examples 61
6.4.1 Joke 61
6.4.2 Fede 62
6.4.3 Nils 62
6.4.4 DLS 63

7 Summary **64**

Bibliography **66**

Abstract

The exchange of geometrical information between different kinds of CAD systems has always been a problem. They have their own internal file formats which are native for every CAD system. Usually a CAD system has different interfaces, but the subset supported of one standard can be different in other CAD systems.

This thesis deals with the problems of the exchange of geometrical information and presents the different ways to represent three dimensional geometry. The emphasis is on the representation of solid models which is a non-ambiguous way to describe three dimensional geometry.

The purpose of the work has been to develop and implement an interface for transferring a three dimensional geometries between a physics simulation system, called GEANT and different kinds of CAD systems. The transfer of the geometry has been implemented using a French standard, called SET, which is a neutral file format for transferring a three dimensional geometry with its creation history. At present (NOV 91) the interface is implemented only to transfer geometry from GEANT to CAD systems, but the intention is to develop a two-way interface.

Since every CAD company has its own financial interests, the interfaces in different kinds of CAD systems are often implemented poorly. The work required in order to exchange geometrical information between different CAD systems is now very difficult and the lack of proper standards is making it even more difficult.

Appendices

A The system shapes of GEANT

A-1 BOX, a box
A-2 TRD1, a trapezoid
A-3 TRD2, a trapezoid
A-4 TRAP, a general trapezoid
A-5 TUBE, a tube
A-6 TUBS, a segment of a tube
A-7 CONE, a conical tube
A-8 CONS, a segment of a conical tube
A-9 SPHE, a segment of a spherical shell
A-10 PARA, a parallelpiped
A-11 PGON, a polygon
A-12 PCON, a polycone
A-13 ELTU, a cylinder with an elliptical section
A-14 HYPE, a hyperbolic tube
A-15 CTUB, a cut tube
A-16 GTRA, a twisted trapezoid

B Some subroutines of GEANT-SET interface

- **B-1** GTOSET, the main subroutine which decodes the GEANT data structure
- **B-2** BOXSET, the subroutine which converts a box into SET
- **B-3** PGOSET, the subroutine which converts a polygon into SET
- **B-4** POSIT, the subroutine which positions and rotates volumes
- **B-5** TREE, the subroutine which writes the physical tree structure of GEANT into SET

C The description of a test geometry, called Joke

- **C-1** Joke drawn by GEANT
- **C-2** Joke drawn by EUCLID
- **C-3** The logical tree structure of Joke in GEANT
- **C-4** The physical tree structure of Joke in EUCLID
- **C-5** The source code for creating Joke in GEANT
- **C-6** The SET file produced by the GEANT-SET interface of Joke

D The description of a test geometry, called Fede

- **D-1** Fede drawn by GEANT
- **D-2** Fede drawn by EUCLID
- **D-3** The logical tree structure of Fede in GEANT
- **D-4** The physical tree structure of Fede in EUCLID
- **D-5** The source code for creating Fede in GEANT
- **D-6** The SET file produced by the GEANT-SET interface of Fede

E The description of a test geometry, called Nils

- **E-1** Nils drawn by GEANT
- **E-2** Nils drawn by EUCLID
- **E-3** The logical tree structure of Nils in GEANT
- **E-4** The source code for creating Nils in GEANT
- **E-5** The SET file produced by the GEANT-SET interface of Nils

F The description of a test geometry, called DLS (Dilepton Spectrometer), a part of a real detector.

- **F-1** DLS drawn by GEANT
- **F-2** DLS drawn by EUCLID
- **F-3** The logical tree structure of DLS in GEANT
- **F-4** The source code for creating DLS in GEANT
- **F-5** The SET file produced by the GEANT-SET interface of DLS

Glossary

3-D Three dimensional.

AeroSpatiale A large French aerospace company which was behind setting up the SET standard.

ANSI American National Standardization Institution.

AUTOCAD Mainly PC-based CAD and drawing package, in widespread use. Produced by Autodesk.

B-Rep Boundary Representation. A solid is represented as the volume contained in a set of surfaces.

B-spline A method to express curves

CAD Computer Aided Design.

CADD Computer Aided Detector Design.

CAM Computer Aided Manufacturing.

CSG Constructive Solid Geometry: solids are described as combinations of simpler solids (primitives) in a series of Boolean logical operations. Standard primitives are cone, cylinder, sphere, torus, block and right angular wedge. Operations are union, intersection and difference. Other CSG primitives are swept solids (of revolution or linear extrusion of a planar face - which

may contain holes) or half spaces (the "semi"-infinite solid one one side of a surface). Composite CSG solids may also participate in logical operations.

CATIA A solid modelling system produced by Dassault Systems.

DLS Dilepton spectrometer. A detector used in Lawrence Berkeley Laboratory experiment in Berkeley.

EUCLID A solid modeller produced in Lyon and developed through the 80s by MATRA DATAVISION (MDTV). EUCLID-IS is an interactive CAD system based on EUCLID and was used in the design and construction of LEP. CERN has thus a large investment in data in the EUCLID format. Aware that EUCLID is aging, MDTV have plans to replace it by a new product [TOP GUN] in the next few years. TOP GUN will have an in-house modeller, rather than adopting ACIS or other available modeller.

EXPRESS a compilable formal Data Modelling Language adopted by the ISO working group on STEP to specify STEP entities and help to ensure the correctness of their use. It contains declarative statements, executable statements and algorithms. It is an ISO Standard: ISO 10303-11, Part 11: Data Modelling Language Express.

GEANT This suite of programs simulates events in the detector. The detector is described by the physicist as a collection of volumes in different materials. Particles are tracked through these volumes, taking into account their identity, energy, the material of the volume, magnetic fields etc. to check that the detector is capable of measuring the required characteristics of the event. As millions of tracks are examined the tracking code must be efficient. GEANT uses sixteen basic forms, combinations of these forms describe the detector volumes and highly optimized code calculates whether a particle is in a given volume, how far it is from the volume surface etc. as the particle

is followed. The description of the detector is close to the CSG description used in some CAD solid modellers and in principle GEANT should be able to exchange data with these CAD systems. CADD is writing an interface between GEANT and the SET standard to allow the exchange of models with CAD systems. Eventually the STEP standard will be adopted to replace SET.

GOSET a French company formed to promote the use and harmonious development of the SET standard.

HEP High Energy Physics.

I-DEAS CAD/CAM system produced by SDRC (Structural Dynamics Research Corporation).

IGES Initial Graphics Exchange Specification. ANSI standard for the exchange of product data. Version 1 was in 1980, the current version is 5, but the most popularly available version is 4.

ISO International Standardization Organization.

JVOLUM Data structure for describe geometry in GEANT.

L3 A detector in the LEP beam.

LEP Large Electron-Positron collider

LHC Large Hadron Collider

NFF Neutral File Format. A system-independent common file format for exchange of (CAD/CAM) data.

PS Proton Synchrotron accelerator.

SET Standard d'Echange et Transfert. French national standard (AFNOR) for the exchange of product data. Version 1 concocted by Aerospatiale (1984) version 2 (1989).

SPS Super Proton Synchrotron accelerator.

STEP Standard for the Exchange of Product model data. The forthcoming ISO standard for industrial design and manufacture. Work started in 1983 and it is expected, and has been for a few years, that the first part of the standard will be published this year. The first part will concern the Sculptured Surface and 2-D representations since these are the most used and best understood. Solid representation specifications should be published next year.

Surface model Definition of a geometric model by means of simple and composite surfaces.

VDA Verband der Deutsche Automobilindustrie - the German Association of Car Manufacturers.

Wire-frame Edge representation of 3-D objects, consisting of curves and lines.

YSET SET file analyzer produced by Association GOSET.

Chapter 1

Introduction

1.1 Background

The detectors that are being proposed for experiments at the Large Hadron Collider (LHC) at CERN are very complex devices from an engineering point of view. Designing and building, and being able to use these detectors within a period of five to eight years presents a challenge that can only be met by using sophisticated Computer Aided Detector Design (CADD) and engineering tools in close co-operation with the physics simulation and data analysis environment (GEANT).

The principal applications of GEANT are the tracking of particles through an experimental setup e.g. for acceptance studies and the graphical representation of the setup and of the particle trajectories.

To face this challenge we have to take advantage of all the resources available in the European HEP community. It is very important to give all these collaborators the possibility to participate in the way they like best, know best, and can best. This implies that it must be possible for them to use the design tools in use at their site.

For this reason CERN has started the CADD initiative at the beginning of 1991. The goal of this initiative is to create a sys-

tem which allows free communication between GEANT and different kind of CAD-systems and between CAD systems themselves.

CADD has decided to use initially a neutral file format called SET (Standard d'Echange et de Transfert) for transferring geometrical information. Actually this seems to be the only neutral file format suitable to our needs which is a standard and which has capabilities to transfer large and complex 3-dimensional solid geometry. In future SET is maybe going to be replaced with STEP (Standard for the Exchange of Product model data).

The work leading to this thesis started at the end of March 1991. This work is complicated by the fact that we need to go into exchange of 3-D geometrical information very deeply. If we are successful, the knowledge and the experience gained with this project will be available for everyone and aid the design of the HEP detectors.

1.2 CERN

CERN (Conseil Europeen pour la Recherche Nucleare, European Organization for Nuclear Research), is one of the largest research centers in the world. Established in 1954, the laboratories are located at the French-Swiss border, just outside Geneva.

The convention of CERN states:

The Organization shall provide for collaboration among European states in nuclear research of a pure scientific and fundamental character, and in research essentially related thereto. The Organization shall have no concern with work for military requirements, and the results of its experimental and theoretical work shall be published or otherwise made generally available.

CERN has some of the largest scientific machines in the world, which make of it the world's leading particle physics research institution. Experiments at CERN reach the far frontiers of basic physics, exploring the microstructure of matter deep within the atomic nucleus. This research has revealed many new aspects of subnuclear

behavior, and has led to new insight in the basic forces of nature. Physicists at CERN recreate conditions similar to those at the very beginning of our Universe. This gives also valuable input to astrophysics, in the understanding of the Big Bang. [4]

1.2.1 Organization

The Organization has currently 16 member states, Austria, Belgium, Denmark Finland, France, Germany, Greece, Italy, the Netherlands, Norway, Poland, Portugal, Spain, Sweden, Switzerland and the United Kingdom. The member states contribute to CERN's budget (843 million Swiss francs in 1990) in proportion to their national revenues. The member states also provide the CERN staff, as well as technical expertise in the experiments themselves, or in related machinery. The Council, composed by two delegates from each member state, is the governing body of CERN. Figure 1.1 shows the governing structure of CERN.

To design and build the intricate equipment at CERN, and to ensure its smooth operation, to help prepare, run, analyze and interpret the scientific experiments, as well as carry out all other kinds of tasks in such a large organization, CERN employs more than 3000 people, comprising a wide range of personnel, physicists, engineers, technicians, administrators, workmen etc. These employees are distributed across 10 divisions. The divisions are composed of groups, which again may contain several sections. In addition to the staff, CERN attracts scientists from the member states all of the world. (Most of them are physicists, paid by their home universities.) This brings the total number of people working at CERN up to 8700 (1990). The internal organization of CERN is shown in figure 1.2. [4]

The work leading to this thesis is carried out in Application Software Group (AS) of Computing and Networks Division (CN).

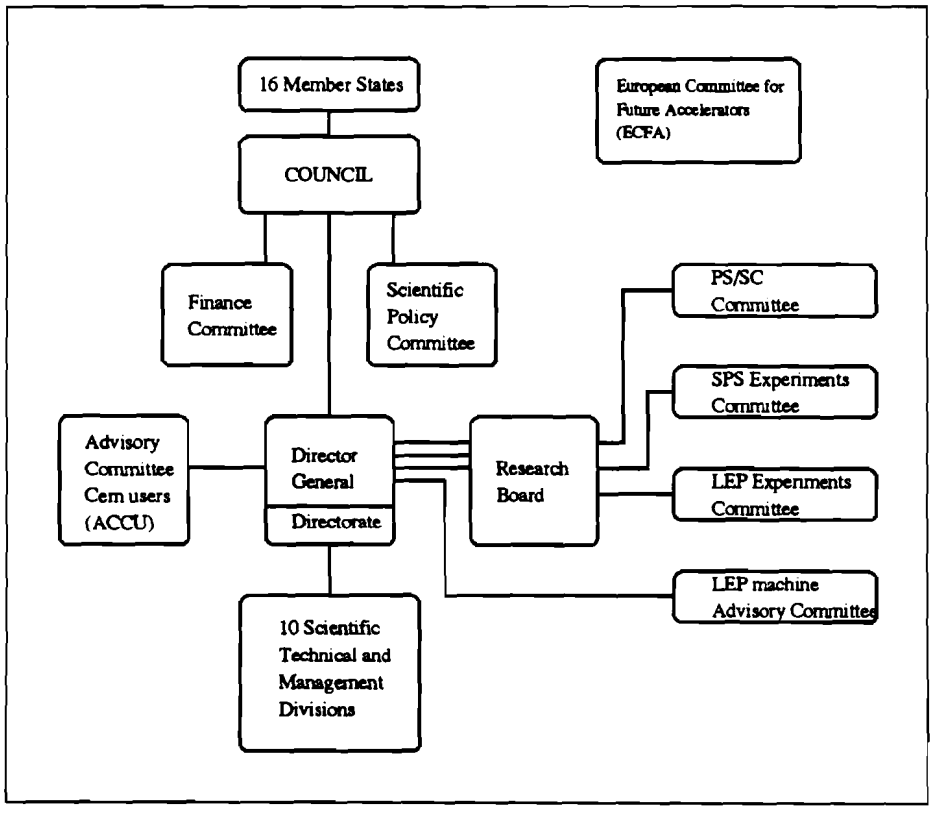


Figure 1.1: The governing structure of CERN

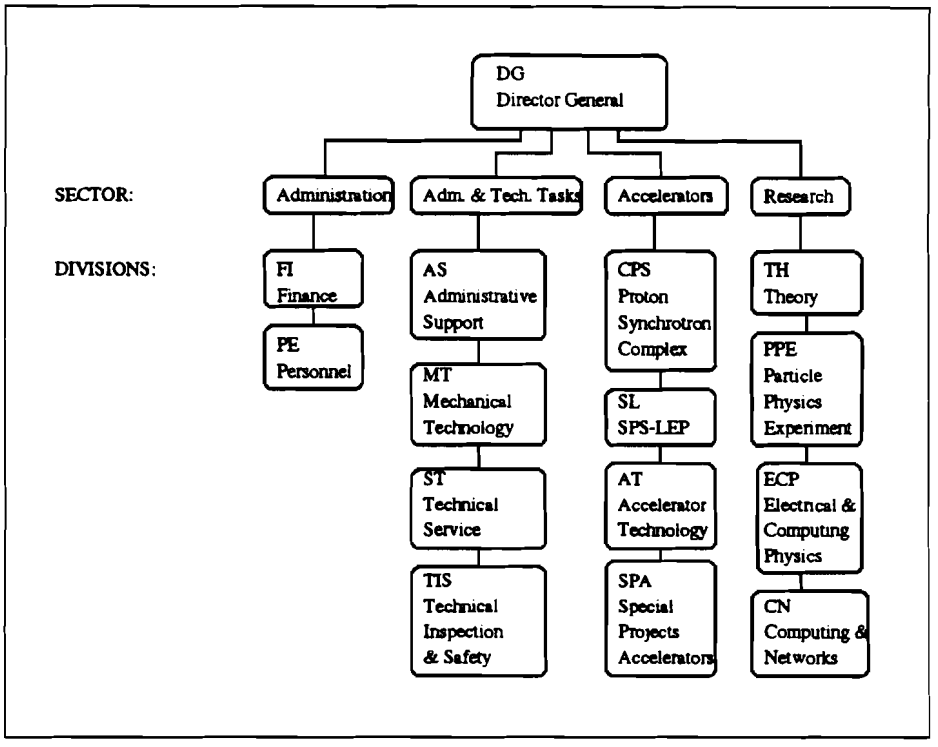


Figure 1.2: The internal organization of CERN

1.3 Detectors

The central equipment at CERN are the accelerators, ranging from the Synchro-Cyclotron, SC, first start up in 1957, able to give protons an energy of 0.6 GeV, to the Super Proton-Synchrotron, start up 1976, capable of giving energy levels of 450 GeV, and the gigantic 27 km circumference Large Electron-Positron Collider, LEP, finished in 1989. These machines are the main worktools required for the physics experiments. High energy beams are here turned into collisions, which generate new unstable particles as energy turns into matter. The results are then observed in surrounding detectors. In figure 1.3 is shown an exploded view of a detector, called Aleph.

A particle accelerator in principle consists of an evacuated tube, surrounded by a magnetic field. (Similar to a Cathode-Ray tube in a television or a computer screen.) To reach energy levels necessary for the experiments, one has to apply the accelerating magnetic field for a long distance. The most convenient way to do this is to let the particles pass the fields several times, i.e. keeping them in a closed path. This is done by means of another magnetic field, forcing the charged particles to follow a circular path. Because of the limitations of magnet technology, one has to increase the radius of the accelerator circle, in order to reach the desirable energy levels.

Thus the SPS has a diameter of 2.2km. Re-designed between 1978-81 as as a proton antiproton collider, it produces energy levels of 630 GeV in the center of mass. Fed from the smaller PS accelerator, which acts as an injector, the use of SPS as a collider has led to the discoveries of the W and Z bosons. This earned the Nobel Prize for Physics to Carlo Rubbia and Simon van der Meer.

The LEP, located in an underground tunnel beneath the Swiss and French countryside, is CERN's latest major engineering achievement. Using the PS and SPS as injectors, it allows for experiments with collisions between electrons and positrons at energy levels of 110 GeV in the center of mass.

CERN's next big project, the Large Hadron Collider, LHC, is

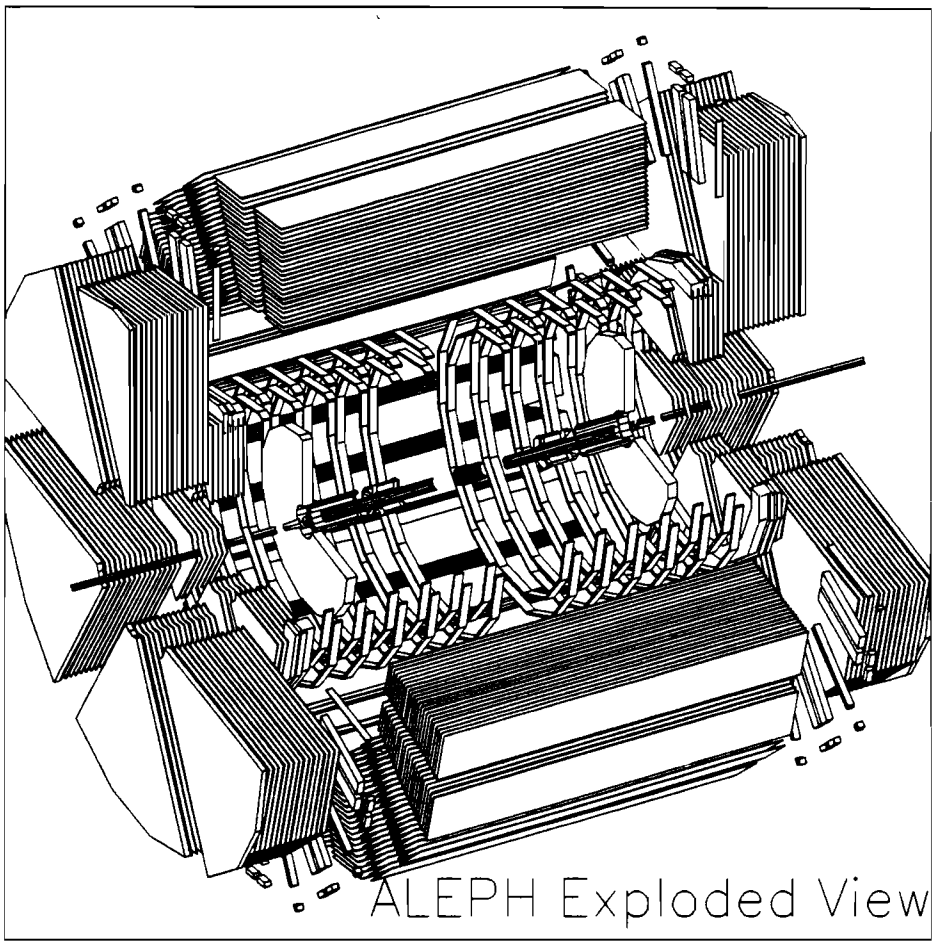


Figure 1.3: An exploded view of Aleph drawn by GEANT

an accelerator for proton-proton collisions at energy levels of 16000 GeV in center of mass. If approved, the LHC will be built on top of the LEP, utilizing the same tunnel. The desirable energy levels should be reached by means of superconductive magnets. The LHC project gives new challenging engineering tasks, mainly in advanced technologies related to the magnets and detectors.

One of the Europe's largest and most powerful computer center provides the computing power for the tremendous amount of analyses, simulations, control systems and all other kinds of calculations and informatics used in such an organization. Efficient network facilities inside the organization and with the rest of the world, allows for the communication among different institutes. The scientific work done at CERN, making stringent demands, frequently pushes existing technology to its limit, which again leads to innovation, with spin-offs in other fields, stimulating European industry. [4]

1.4 Purpose of the work

Mechanical CAD/CAM is, or will be, used by all the HEP detector builders in Europe. As experiments become bigger and technically more advanced, building a detector becomes a more complex project. Designing and manufacturing different parts of a detector in different places in Europe must be coordinated to ensure, before building starts, that the elements may be assembled and that they will function properly when assembled together.

The engineering descriptions of elements have to be available for designers and the designer of one element should have access to the data of neighboring elements.

The geometrical description of a detector is first designed by physicists. Physicists are using GEANT to build the best suitable detector for their purposes. The description of the detector must be transferred to a CAD system for engineer design. Of course there is no support elements and engineers must design a detector which can also be built. If the physicists want to know how the support

structures will affect the simulation result. So the geometrical description of the detector has to be transferred back to GEANT. If the subsequent results of the simulation is imperfect, the physicists must change the geometry. This cycle continues until the physicists and engineers are satisfied. [5]

1.4.1 The goal

The goal of CADD is to create a system which allows free communication between GEANT and different kind of CAD-systems used to design LHC detectors. The concept of CADD is shown in figure 1.4. HEP event simulation tools are not numerous, but their evolution is directly influenced by the needs of the HEP community. GEANT, the primary simulation tool considered for use in CADD, will have a new detector geometry database implemented with the aim of easing communication with CAD systems. In GEANT the detector is described by the physicists as an assembly of volumes filled with material. Particles from simulated events are tracked through these volumes to examine the response of the detector. As millions of tracks are followed at each collision, the tracking code must be very fast. GEANT uses sixteen basic geometrical shapes, combinations of these shapes describe the detector volumes and highly optimized code calculates whether a particle is in a given volume, how far it is from the volume surface etc as the particle is translated. The sixteen shapes were chosen because they were useful in describing detectors and because they enable fast tracking code. Such a description of the detector is close to the Constructive Solid Geometry (CSG), description used in some CAD solid modellers.

In principle GEANT should be able to exchange data with these CAD systems. This is desirable, not only to ensure the accuracy which otherwise may be lost in manual transcriptions but also because the amount of data can be very large - there are over 1,200,000 volumes in the GEANT model of L3, which is one of the large detectors installed in the LEP beam.

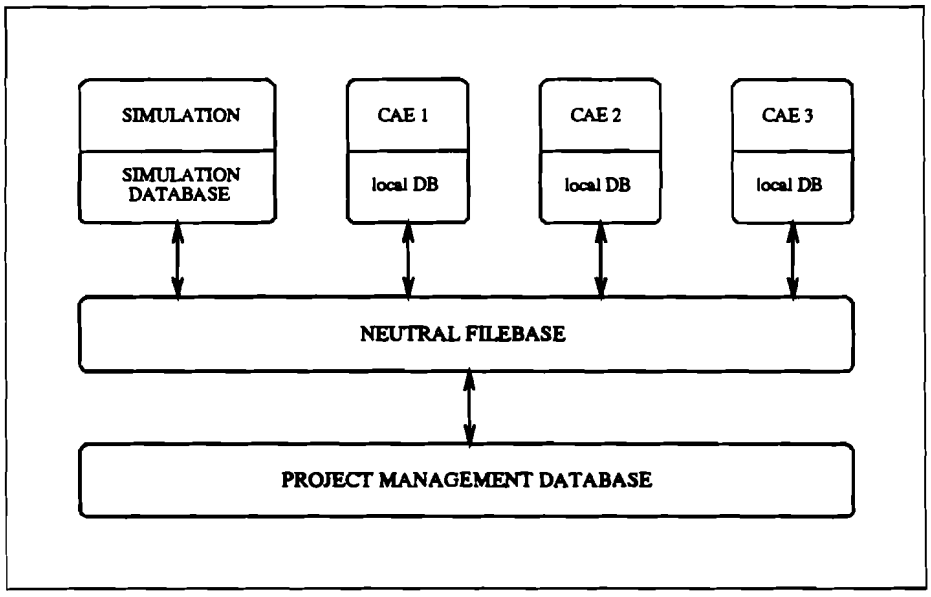


Figure 1.4: The concept of CADD

Work on GEANT is continuing and a new implementation of tracking - this time based on a surface description of the detector volumes - may be running by the end of 1992. These new volume descriptions would free the physicists from the limitations of the sixteen shapes, while having them available as "surface-macros" where convenient. The surface descriptions of volumes correspond to another style of CAD model representation, Boundary Representation (B-rep).

To ensure the coherence of the different aspects of a detector, we need a common dataset describing all equipment relevant to any given detector. This dataset should allow complete information exchange from design to installation and be accessible to the software of the detector teams in Europe. During the design phase the master design file for a detector element will be kept in the institute which is working on that element.

For financial or technical reasons, LHC partners have different CAD tools. They prevent CADD from imposing a unique CAD system on the partners. Besides, it is impossible to predict which, if any, of the currently available systems will exist a decade from now. Thus there is the problem of communicating CAD data from one system, with its own geometrical modeller and its own techniques, to another. One might have a direct interface between each pair of systems or an interface between each CAD system and a standard file format. [5]

1.4.2 History

CADD was launched at CERN in December 1990. It was welcomed as an exercise which would have to be done and which, if it could be in operation in time for the design of LHC detectors, would bring earlier benefits. Indeed, the countback from installing experiments in LHC in 1997 requires that CADD must be running in 1992 and be strong and reliable in 1993.

It was decided to use Neutral Format computer Files (NFF) as

the tool through which CAD systems would exchange data since we could not select a single CAD system writing interfaces from GEANT to each of the CAD systems used or to be used. NFF are used in large industrial concerns like AeroSpatiale and the German Association of Car Manufacturers (VDA).

Lacking direct experience in this field it was decided to begin an investigative implementation of a CADD dataset to find out what the problems might be. The French standard SET was chosen as Neutral File Format for this experiment since SET appeared more advanced than the ANSI standard IGES. In particular, SET provided a means of expressing geometrical entities in CSG which is close to the means used in the present GEANT geometry database to describe detector elements - and the inclusion of GEANT in the design cycle is a kingpin of the CADD idea. [5]

1.4.3 Problems

Problems are of at least three sorts.

1. Engineering standards are large and complex thus expensive to implement. So a CAD vendor tends to implement only that part of a standard which is required by his most important customers. The result is that the implementation of a standard provided by one vendor may have little in common with the implementation of the same standard by another vendor. Hence the exchange of data between the two systems is restricted.

2. Since the current standards - IGES and SET are the principal ones - were issued, there has been a growing awareness that it is not enough to define entities in a standard, but there also must be rules for their use, and ways of enforcing these rules. In other words a dictionary does not describe a language, there must exist a grammar, syntax, usage and all the other elements which allow people to communicate clearly and precisely. Each CAD system is, after all, an independent creation with its distinct interpretation of the world,

including exchange standards. AUTOCAD, for instance, is an unexpensive and widely used CAD system and it expresses a point in space in the IGES format as a line starting at that point, having zero length. EUCLID, the CAD system used in the construction of LEP, has an IGES interface and on reading AUTOCAD's description of a point in space, dismisses it as incomprehensible.

3. The very nature of a CAD system may be such that it simply cannot cope with certain kinds of geometry. A CAD system made for styling motor car bodies might happily handle attractive curved surfaces but be unable to deal with the elements of the engine.

It is intended that the forthcoming ISO standard, STEP, be adopted in the future, though we do not expect that STEP implementations will be common enough to be useful to CADD before, say, 1994 - too late for early work on LHC.

With the STEP standard the exchange problem will decrease since hundreds of man-years have been spent in making STEP more complete than its predecessors. STEP will embody, notably, application protocols which define exactly what entities will be expressed in a given field of work, and how the CAD system should express them. STEP entity definitions are written in a formal data description language called EXPRESS. Meanwhile CADD faces the problem of making different CAD systems communicate in the currently defined standards.

The EUCLID interface with SET is fairly complete and will probably be able to read the design data from GEANT. However the SET interfaces of I-DEAS and Pro/Engineer, say, do not cover enough of the SET standard to read these data. Indeed the situation of SET interfaces is similar to that of IGES interfaces. Given that the CAD world is waiting for the STEP standard, it is most unlikely that the CAD vendors can be convinced to devote resources to further development of their interfaces to standards which are getting out-of-date.

There are many elements in the CADD. Some, like networks and workstations will develop independently to improve performance. The state of exchange interfaces is clearly unsatisfactory in general but these can still be used to build motor cars and aeroplanes and it is up to CADD to avoid the difficulties in this field.

This requires detailed work. STEP standard will improve the situation and the effort invested in using the current standards will help to understand better the different natures of CAD systems and how they, and the different engineering traditions of Europe, may be connected. It seems unlikely that all of the elements of a full CADD system will be ready in time for early work on LHC detectors, but those parts of CADD which are working will probably help to make better detectors. [5]

Chapter 2

GEANT — detector description and simulation package

2.1 General

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

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

GEANT is a system of detector description and simulation tools which should aid physics for such studies. The first version of GEANT was written in 1974 as a bare framework which initially emphasized tracking of either one or a few particles per event through relatively simple detectors. The system has been developed with some continuity over the years.

The principal applications of GEANT are

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

In view of these applications, the GEANT system allows

- to describe an experimental setup in an efficient and simple way. The setup is represented by a structure of geometrical volumes. Each volume is given a medium number by the user. Different volumes may have the same medium number. A medium is defined by a set of parameters, the so-called tracking medium parameters, which include reference to the material filling the volume.
- to generate simulated events from standard HEP Monte Carlo generators. (Monte Carlo is a common method used in most simulation applications in the world)
- to control the transport of particles through the various regions of the setup, taking into account the geometrical volume boundaries and all physical effects due to the nature of the particles themselves, to their interactions with the matter and to the magnetic field.
- to record the elements of the particle trajectories and the response from the sensitive detectors.
- to visualize the detectors and the particle trajectories.

Part of the subroutines available in GEANT are integrated into program segments prepared for the execution of some of these tasks. The program segments may contain 'dummy' and 'default' user subroutines called whenever application dependent actions are expected. The remaining subroutines are tools either to perform simple functions or to implement the non trivial operations required for most of the applications. [1]

2.2 The geometrical modeller of GEANT

The geometry package of GEANT consists of subroutines which can be used in the initialization phase of the program to describe the geometry of the experimental set-up, and of subroutines which ensure the communication with the tracking package during the event processing phase.

2.2.1 The volume definition

Experimental set-ups, as complex as the detectors prepared for LEP and in future for LHC, can be described rather accurately through the definition of a set of simple volumes.

Each volume is given a name and is characterized by:

- A shape identifier, specifying one of the basic geometrical shapes available
- The shape parameters, giving the dimensions of the volume
- A local reference system, whose origin and axes are the ones defined for the given shape.
- Its physical properties, given by a set of constants for the homogeneous material which fills the volume
- Additional properties, known as 'tracking medium' constants, which depend on the characteristics of the volume itself and on its geometrical and physical environment
- A set of attributes, in connection with the drawing package and the detector response package

As long as it is not 'positioned' in a given reference frame, a volume is an entity which has no spatial relation with the set-up. By convention, an initial volume has to be defined which should match (or surround) the outside boundaries of the entire set-up. The reference frame attached to this volume is considered to be the master reference frame. [1]

2.2.2 Basic shapes

GEANT has 16 basic system shapes which can be found in appendix A. The system shapes supported at present are as follows [1] :

- **BOX** is a box. It has 3 parameters, the half lengths in x, y and z.
- **TRD1** is a trapezoid with only the x length varying with z. It has 4 parameters, the half length in x at the low z surface, that at the high z surface, the half length in y and in z.
- **TRD2** is a trapezoid with both x and y lengths varying with z. It has 5 parameters, the half length in x at the low z surface, that at the high z surface, the half length in y at the low z surface, that at the high z surface, the half length in z.
- **TRAP** is a general trapezoid, i.e. one for which the faces perpendicular to z are trapezia and their centers are not at the same x, y. It has 11 parameters: Dz the half length in z, Th & Phi the polar angles from the center of the face at z=-Dz to that at z=+Dz, H1 the half length in y at z=-Dz, LO1 the half length in x at z=-Dz and y=low edge, LH1 the half length in x at z=-Dz and y= high edge, Th1 the angle w.r.t. the y axis from the centre of the low y edge to the centre of the high y edge, and H2, LB2, LH2, Th2 the corresponding quantities to the 1s but at z=+Dz.
- **TUBE** is a tube. It has 3 parameters, the inside radius, the outside radius and the half length in z.
- **TUBS** is a phi segment of a tube. It has 5 parameters, the same 3 as 'TUBE' plus the phi limits. The segment starts at the first limit and includes increasing phi value up to the second limit or that plus 360 degrees.
- **CONE** is a conical tube. It has 5 parameters, the half length in z, the inside and outside radii at the low z limit and those at the high z limit.

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

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

- **PARA** is a parallelpiped. It has 6 parameters, the half length in x, the half length in y, the half length in z, the angle w.r.t. the y axis from the centre of the low y edge to the centre of the high y edge, and the theta phi polar angles from the centre of the low z face to the centre of the high z face.

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

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

- **ELTU** is a cylinder with an elliptical section. It has three parameters: the ellipse semi-axis in X, the ellipse semi-axis in Y and the half length in Z. ELTU is not divisible.

- **HYPE** is a hyperbolic tube, ie the inner and outer surfaces are hyperboloids, as would be formed by a system of cylindrical wires which were then rotated tangentially about their centers.

The 4 parameters are the inner and outer radii, the half length in z, and the "stereo angle" theta in degrees.

- **CTUB** is a cut tube with 11 parameters.
- **GTRA** is a general twisted trapezoid with 12 parameters.

2.2.3 Creation of shapes

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

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

Divisions can be performed along any of the three axes of the mother volume. The definition of the axes depends on the shape. A volume can be partially or totally divided. The division generates a cell, which is considered as a new volume which is one of the allowed shapes. Its dimensions are computed according to the declared division number and/or step size. This cell, as any volume, can again be divided along any of its proper axes, or have other volumes positioned into it. Volumes positioned within a cell are reproduced implicitly in all cells.

These operations permit to define a physical tree of volumes at different levels. It is assumed that the 'tracking medium' properties of the contents replace the ones of the mother within the space region they occupy. A volume is therefore defined not only by its intrinsic characteristics but also by the definition of its 'descendants', namely its contents, the contents of its contents, etc.

The package accepts a maximum of 15 levels, which should be quite enough to represent even the fine details of a complex set-up. [1]

2.2.4 Data structure

The physical tree is represented in memory by a logical tree structure, the JVOLUM data structure, which describes the arrangement of volumes in a compact and recurrent way, and accepts a maximum of 15 levels. Each 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 volumes corresponding to those contents. The principle of the data structure is shown in figure 2.1.

In the situation where division or multiple copies occur, there is no longer a one-to-one correspondence between a given volume in the logical tree and a unique region in space. Information has to be kept at tracking time to identify which division cell or which copy was considered at each depth level along the path through the physical tree. [1]

The volume data structure JVOLUM contains following variables:

- **ISHAPE** system shape number
- **NIN** number of volumes imbedded in the mother volume. If it is negative then the volume is divided
- **NMED** medium number for the volume
- **NPAR** number of shape parameters
- **NATT** number of other attributes
- **PAR** array of shape parameters
- **ATT** array of other attributes

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

- **IAXIS** defines the direction of the slices (1,2,3)
- **IVO** system volume number
- **NDIV** number of slices
- **C0** minimum coordinate limit
- **STEP** coordinate step from slice to slice

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

- **IVO** system volume number
- **NR** user number
- **IROT** rotation matrix number defining the orientation of the volume
- **X,Y,Z** define the position of the volume
- **KONLY** indicates whether it is sufficient to find a point within this volume or whether there may be some ambiguity with other volumes

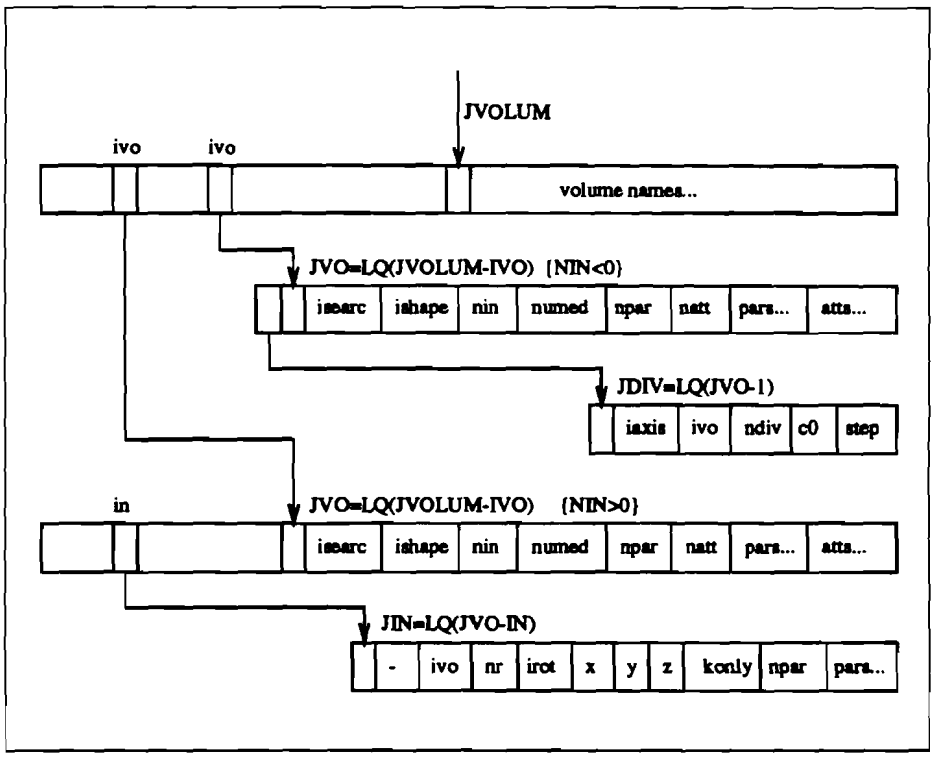


Figure 2.1: The data structure of GEANT

Chapter 3

3-D geometric modelling in CAD

CAD/CAM technology in design and manufacturing is now playing an increasingly important role in production industry. The importance of CAD/CAM has been widely recognized. CAD/CAM technologies make it possible to shorten the time and lower the cost of development. Additionally, the reliability and the quality of the product can be improved. CAD/CAM systems have therefore been used in various fields of industry including automobile and aircraft manufacture, architecture and shipbuilding, and there are currently many commercial CAD/CAM systems available.

One of the problems in CAD/CAM is how to represent information on the product in a computer. In conventional design and manufacturing, product information is described in an engineer drawing. Because the information is described in this form, it can be interpreted only by trained engineers. In the case of CAD/CAM, information must be stored in a computer so that it can manipulated by the user of a computer. Thus the development of techniques for representing and utilizing product information in a computer is very important.

In mechanical engineering, one of the advantages of 3-D CAD/CAM

is the full 3-D description of a model, which can be used as data for workshop machines, visualization in marketing, simplification, etc., and in addition, to represent two-dimensional drawings. A designer will describe the shape of the product in the drawing so that it satisfies various engineering requirements. When using CAD for mechanical design, it is necessary to represent a three-dimensional shape in a computer and therefore techniques are required to construct the representation.

In conventional mechanical CAD systems there is four different way to represent 3-D geometric models:

- Wire-frame models
- Surface models
- Boundary representations
- Constructive solid geometry

A wire-frame model and surface model represents part of a designed object. Boundary representation and constructive solid geometry are solid models and they represent the complete shape of an object as a closed space in 3-D space. Only with a solid model it is possible to check whether or not any point in a space is inside the solid. In reality many CAD systems are hybrid and they use mixtures of boundary representation and constructive solid geometry. [3] [4] [6]

3.1 Wire-frame model

Wire-frame systems were developed in the early 1960s to automate design drafting. Most of the early 3-D CAD systems used wire-frame models. The first systems were only two-dimensional, with the user constructing a model point by point and line by line.

A wire-frame model is represented by tables defining edges and points. The start point and the end point of each edge are stored in the edge table. An edge may be a line or a curve. The coordinates of each point are stored in the point table. This representation is natural for a designer who is familiar with mechanical drawings, since it it is the lines and curves in a drawing which define 3-D shape. A wire-frame model is stored very simply in a computer as a data structure. The storage space is small and the access time very short.

When one attempts to use pure wire-frame representations to model 3-D solids, one is confronted with four problems:

- The possibility of creating ambiguous models.
- The possibility of creating nonsense objects.
- The lack of graphic or visual coherence.
- The possibility that the wire-frame model approximating a solid is verbose.

For simple objects, the the advantages and disadvantages of wire-frame versus solid modeller are not so important as for complex object shapes. As a shape becomes more complex, the level of detail necessary increases rapidly. An ambiguous wire-frame model is shown in figure 3.1 and a non-ambiguous wire-frame model in figure 3.2. [3] [6]

3.2 Surface model

A surface model is represented by tables of edges and points, as is a wire-frame model, but additional to wire-frame, a surface model is represented by tables of faces. The face table stores information on which edges are attached to each face. In most conventional CAD systems for free-form surfaces, surface models have been used

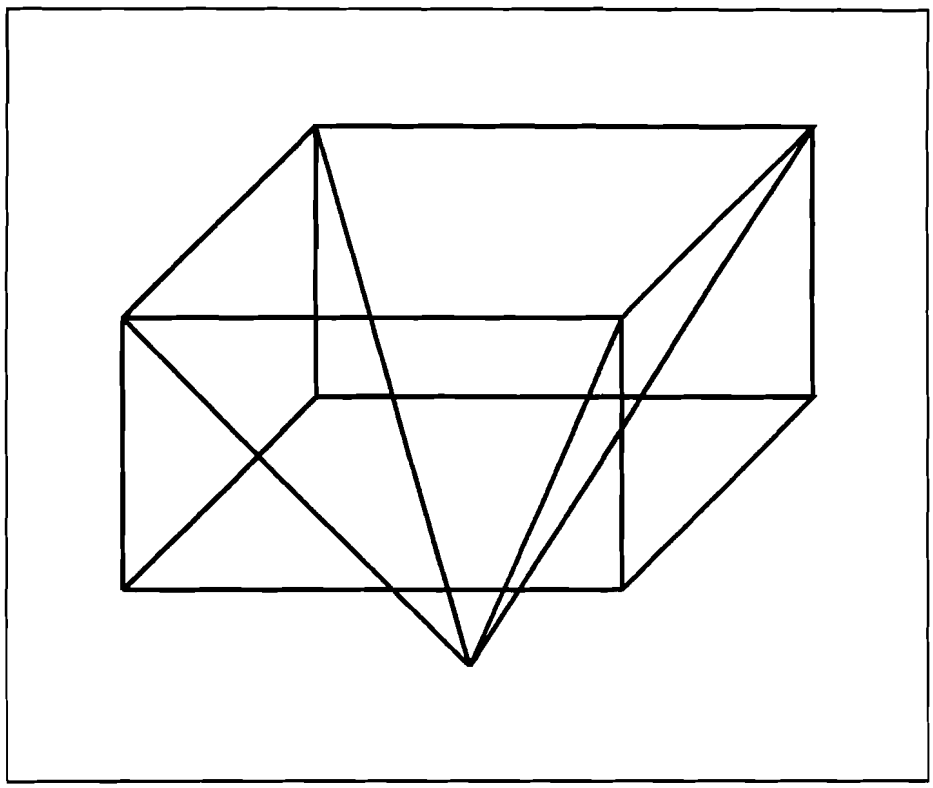


Figure 3.1: An ambiguous wire-frame model

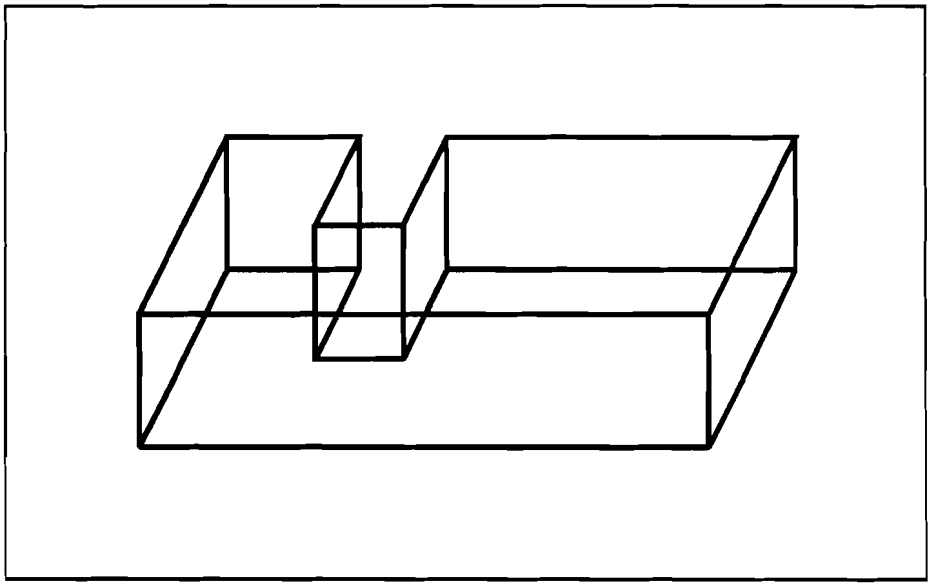


Figure 3.2: A non-ambiguous wire-frame model

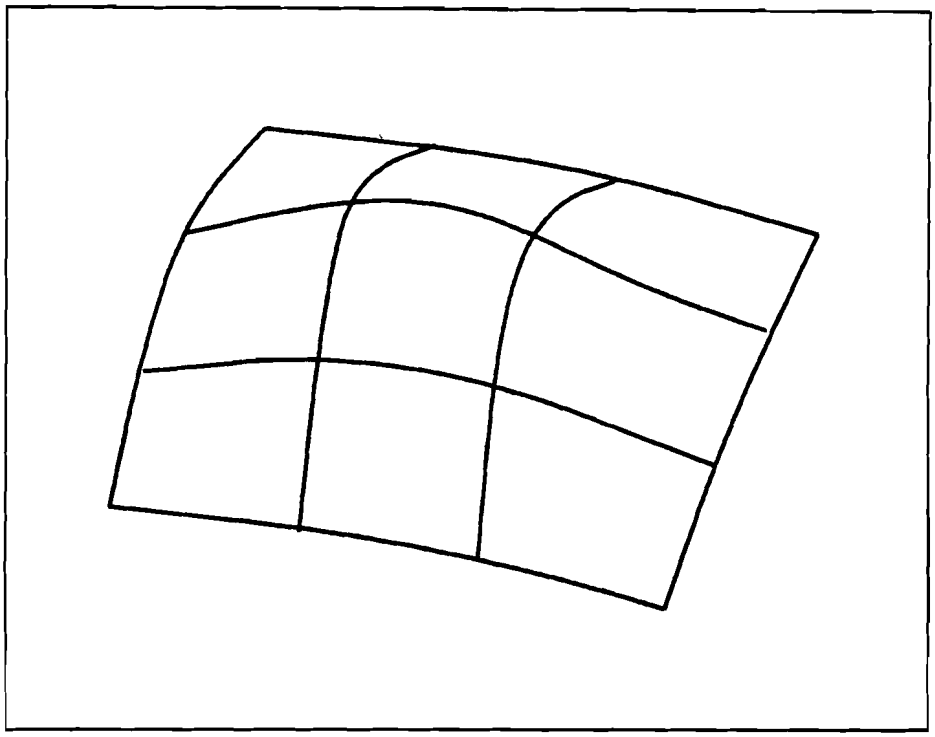


Figure 3.3: A surface model

as internal representations. However, a surface model is a set of faces, and as such is ambiguous when determining the volume of an object. Surface models play an important role in industry, because they give an accurate description of the surface of an object. An example of a surface model is shown in figure 3.3. [3]

3.3 Solid models

Because there is no ambiguity in using a solid model to represent 3-D shape, the importance of using solid modelling in CAD/CAM

has been widely recognized. The solid model can support a wide range of activities, like interference check, computation of volume and surface area, finite element analysis, etc.

Most solid modelling systems have adopted either constructive solid geometry or boundary representation as internal representations of solids. Solid modelling systems will always have some kind of boundary representation for display purposes etc. [3] [6]

3.3.1 Constructive solid geometry (CSG)

In this technique solids are described as combinations of simpler solids (primitives) in a series of Boolean operations. Typical standard primitives are cone, cylinder, sphere, torus, block and right angular wedge. Operations are union, intersection and difference. Other CSG primitives are swept solids (of revolution or linear revolution of a planar face - which may contain holes) or half spaces. Composite CSG solids may also be subjected to logical operations. Using a Boolean operation, a new solid is made from two intersecting solids. This technique is common in mechanical engineering because it gives a precise, analytic description of the body. A simple CSG model with a Boolean operation is shown in figure 3.4.

A body modelled using this technique may be manufactured with high precision, since each step of the process is to subtract (or add) a precisely defined sphere, cylinder etc from the material being worked.

The data structure of constructive solid geometry is simple, and its data size is small. Any system using CSG as an internal representation can provide both a CSG and B-rep, as it generally possible to convert a CSG to B-rep. The system can therefore support a wide variety of applications. Because CSG defines not only the shape of a solid, but also the process of modelling that solid, a modification of the solid is easy.

There are only a limited operations available to create and modify a solid. Generally, it is not easy to implement operations other than

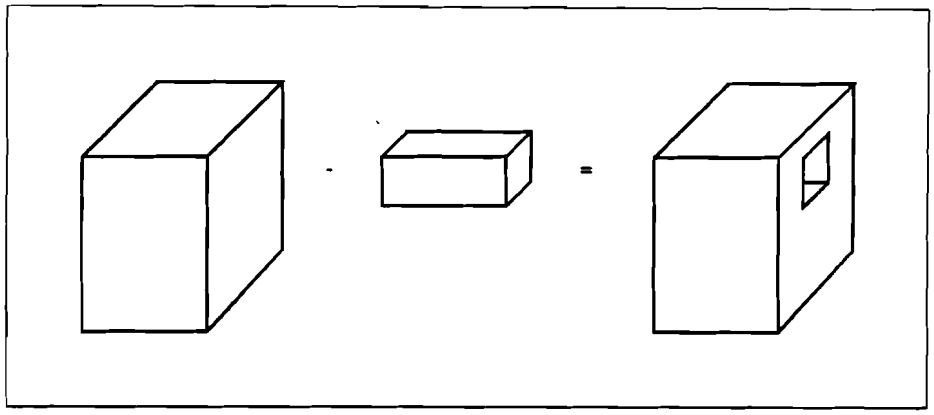


Figure 3.4: A simple CSG model as a result of subtraction

Boolean operations. The computation for generating pictures of solids requires much time. [3] [6]

3.3.2 Boundary representation (B-rep)

A solid is represented as the volume contained in a set of surfaces plus topological information which defines the relationships between the surfaces. Because boundary representation include such topological information, a solid is represented as a closed space in 3-D space. The boundary of a solid separates points inside from points outside the solid. A very simple B-rep model is shown in figure 3.5.

Since surfaces are explicitly represented in a boundary representation, a wire-frame picture of a B-rep is quickly drawn. The data structure of a boundary representation is complex, and it requires a large memory space.

This technique is widely used in the motor car and aeroplane industries since it allows to have descriptions of the surfaces to be used for example in making the presses which are to form the sheets for the wings, doors etc. [3] [6]

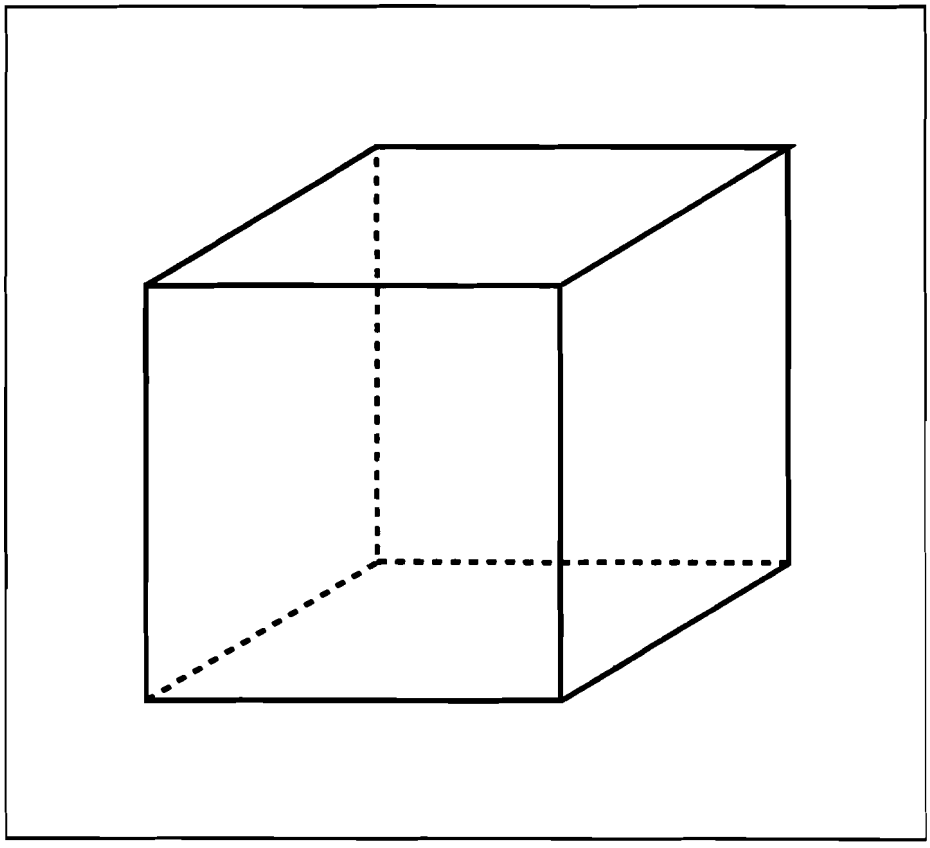


Figure 3.5: A simple B-rep model

Chapter 4

SET —The neutral file format

4.1 General

The Standard for Exchange and Transfer (SET) is the existing French standard for exchange of CAD data. It started as a neutral format made for exchange between different CAD systems at Aerospatiale in 1983. The aim was the need for a reliable alternative to IGES. It became an official French standard, Afnor Z68-300, SET, in 1985. SET was revised and extended in 1989. The current version (89-06), supports wireframe, surface and volume entities. Entities for drafting and connectivity applications, as well as scientific data and FEM modelling are also included.

SET has two main objectives which can be expressed as follows:

- **Exchange** of data between different CAD systems
- **Archiving** of data for later use

Figure 4.1 shows the main objectives of SET.

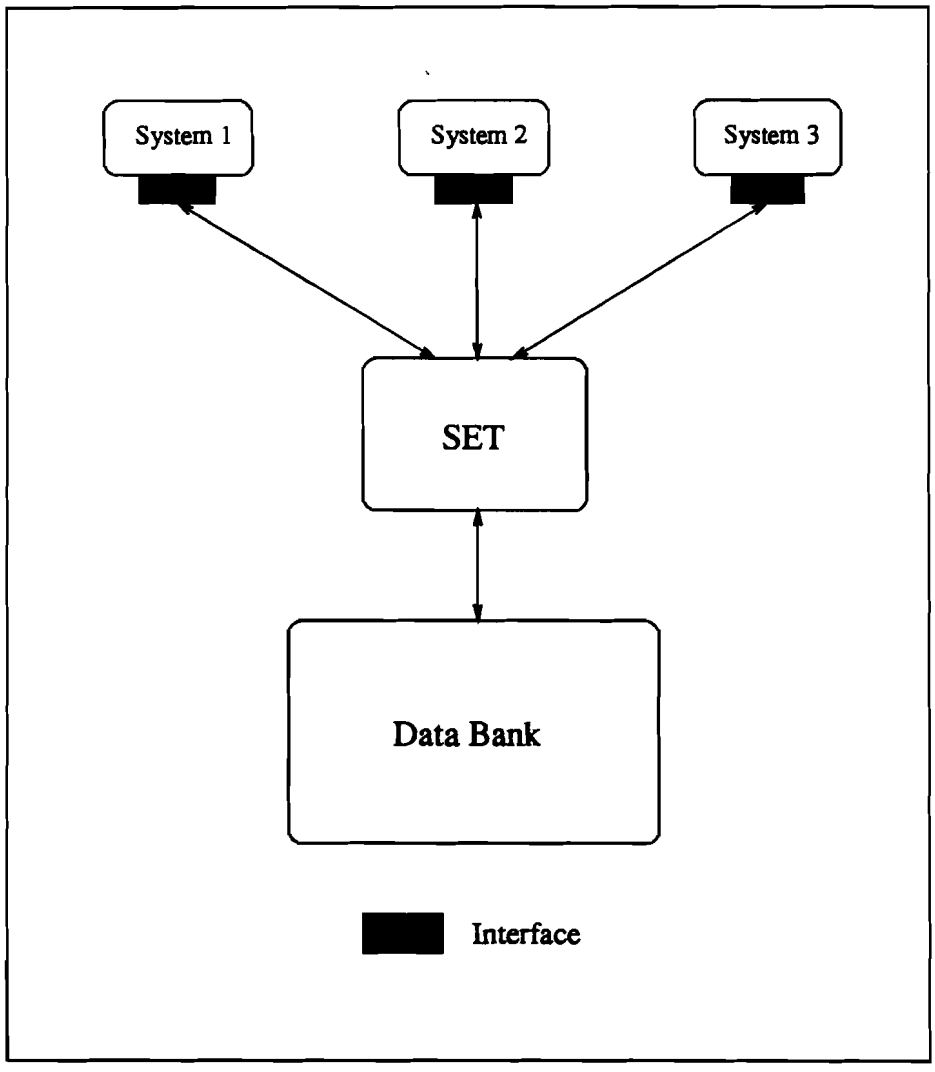


Figure 4.1: Exchange and archiving with SET

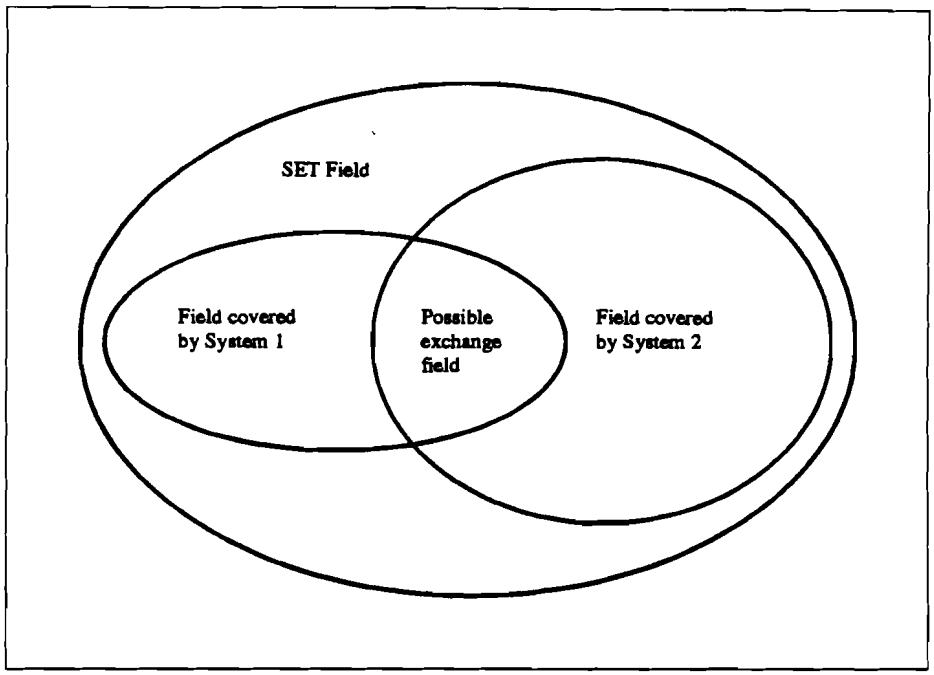


Figure 4.2: Field of exhange and archiving

Because of the different concepts used by the CAD/CAM industry, the standard has to cover a very large field of possible entities. Figure 4.2 shows the different fields of SET.

The philosophy behind the standard is that it should be able to handle 100% of the possible information that may occur (Figure 4.3). It is also considered to be extremely 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. [2] [4]

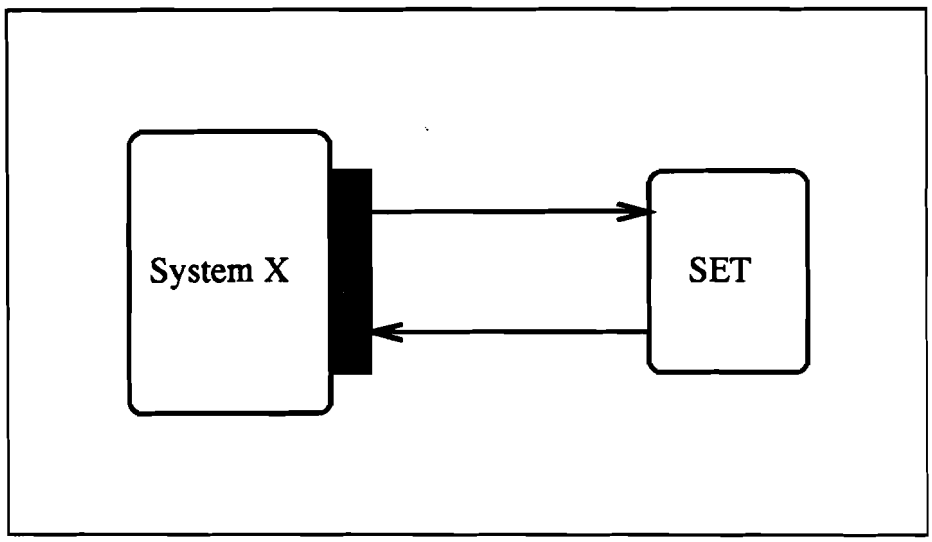


Figure 4.3: 100% information handling

4.2 SET -file format

The structure of SET is based upon a three level hierarchical structure, which consists of data assemblies, data blocks, 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. Start of a SET data assembly is given by block @9901, end of a data assembly by block @9998.
- **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 se-

quence 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.

Examples of blocks are: @2 Line Segment, @41 General B-spline surface, @50 Primitive solid, @701 Finite Element Modelling and @9900 Start of set file.

- **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 the 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.

  Examples of sub-blocks are: #1 Coordinates of a point, #41 Parameters for B-SPLINE surface, #100 Boolean operation and #9901 Date.

- **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.

  Examples of dictionary entries are: :2 Space, :8 layer and :53 Line Style.

References among entities in a SET file are made by use of pointers, either directly from the blocks, or via sub-blocks or dictionary entries. Pointers to other blocks are identified by an !, followed by the blocks (sequence) number.

SET has no unique mapping of types of sub-blocks 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. [2] [4]

4.3 Entity organization

SET blocks and sub-blocks are grouped into classes according to their functional domains. A class is a range of 100 type numbers that is reserved for a domain. These classes are:

- **Class 0:** Basic geometry, like points, curves and surfaces.
- **Class 1:** Constructed or composite geometry, as applicable in the CAD domain.
- **Class 2:** Graphics information, like views and drawings.
- **Class 3:** Geometrical, mathematical and physical attributes.
- **Class 4:** Attributes related to text and dimensioning.
- **Class 5:** Domain for topology and group definitions.
- **Class 6:** Connectivity applications.
- **Class 7:** Finite element related data.
- **Class 80:** Reserved for non standard entities developed by users.
- **Class 99:** SET management. Used for file headers etc.

Dictionary entries are also classified into their field of application:

- General purpose (0-50)
- Relating to representation (50-99)

- Related to text definition (100-1nn)
- Related to graphic symbol definitions (200-2nn)
- Related to Finite Elements (700)

The SET standard also refers to groups of entities in addition to the classes. This is a grouping of the entities into specific areas like curves, surfaces etc. [2] [4]

4.4 The geometrical modelling via SET

In this project, we are most concerned about the possibilities to exchange solid models since the geometrical modeller of GEANT is based on volumes with contents. The most natural way to recreate this consept in a CAD system is to use solid models. 3D models may be represented in several ways in different systems, using different concepts, such as wireframe, CSG, B-rep, etc. These concepts must be combined with the SET structure, to form a reasonable output. Figure 4.4 shows the definition of a solid in SET.

The SET format itself should be suitable for most CAD systems, because of its flexibility. In the Afnor standard Z68-300-8 it is claimed that SET can be used to define purely geometrical entities without reference to any concept of representation whatsoever. Several forms of representation should then be possible. The SET format allows solid entities to be defined in three different ways:

- Constructed by primitives and Boolean operations (CSG)
- Exact boundary representation (B-rep)
- Representation by facets

Combinations and associations between these representation techniques are also possible. This should allow for interfaces between systems using several solid modelling concepts. If enough effort is put into the interface software, one should be able to support most

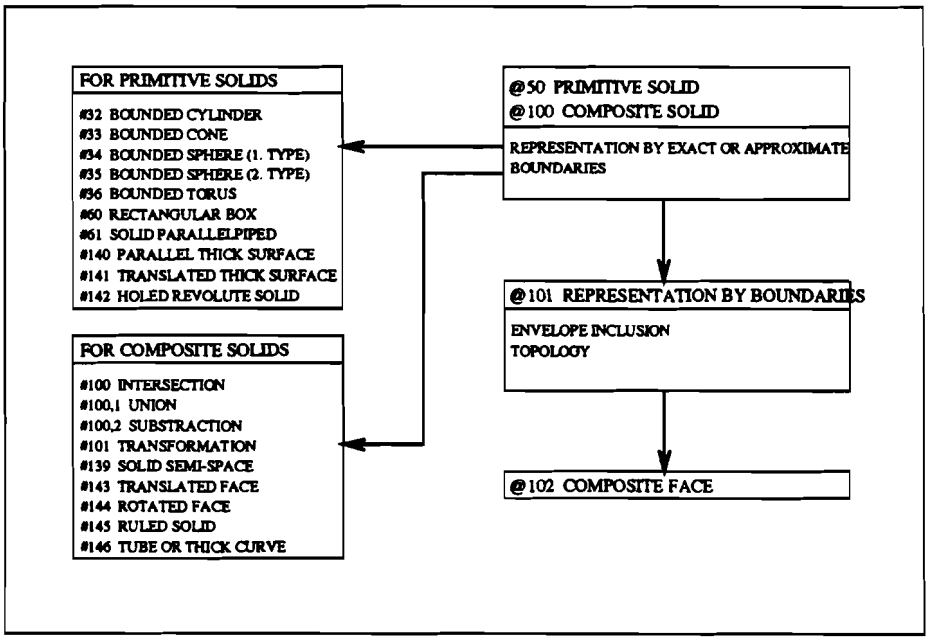


Figure 4.4: Definition of a solid

of the information available from the CAD system in the SET represented object.

Determination of what is volume, and what is empty space, is necessary to define a solid model. In SET, this is done by the usual mechanism of sub-blocks and/or dictionary entries in connection with the relevant blocks. In this case, topological closure is defined by means of dictionary entries. To distinguish the interior of a surface from the exterior, one has mathematical concepts, using normal vector information. For primitive solids, the topology is determined by definition. In other cases (constructed or composite solids), SET uses the concept of envelopes. An envelope is an open or closed boundary surface dividing space into two semi spaces, one consisting of material (interior), and one empty (exterior).

An organization founded by some of the main users of SET, GOSET, follows up the standard, takes care of promotion and further development. For SET, GOSET has a validation service for SET interfaces under development. Even if some vendors are not interested in such activities, hopefully most companies will realize the need for quality control. [2] [4]

Chapter 5

SET - GEANT interface

5.1 General

Work is in progress to implement the interface between GEANT and the SET standard. GEANT shapes written in SET will be read by EUCLID and engineering work done on them. Thus GEANT will be able to communicate its data to one of CADD's principal CAD systems. At present (Oct91) 15 of the 16 solid shapes with which GEANT works can be written to SET and read by EUCLID. The creation history of the geometry can also be written to SET. From EUCLID data may be transmitted to other CAD systems such as I-DEAS but as long as these CAD systems do not have an interface which can handle solid geometry, this will be Wireframe, surface and 2-D data transmitted via IGES, not the original 3-D solid data. Some work will then have to be done to reconstitute the volumes of interest in, say, I-DEAS. CATIA should also be able to read the solid data in the SET format but we have not yet got a CATIA-SET interface.

It is comparatively simple to write an interface from GEANT to SET, this involves clarifying the GEANT data structure, transferring the forms to the SET format and writing out the restricted number of entities and the creation history.

To read a SET file written by a commercial CAD system is more difficult since it is not possible to constrain that system to restrict itself to writing only certain entities and so one must be prepared to deal with the full standard.

As yet no code has been written to allow GEANT to read SET data, but it is intended to use the YSET package, marketed by the GOSET association, to read SET files into GEANT. The YSET package has been at CERN from July 1991 and is being evaluated. [5]

5.2 Environment

The work with the GEANT-SET interface was mostly done with a DECstation 5000/200, which is a powerful workstation. We have disk space over 1.2 Gbytes and memory of 24 Mbytes. This Workstation is used only for developing of the GEANT-SET interface by two users. ULTRIX 4.1 operating system with DECwindows window manager has been a nice tool with some exceptions, which I will describe later in this thesis.

Since GEANT is written in FORTRAN, the work 'had to' be carried out with FORTRAN as well using F77 compiler. The code is produced by A Source Code Management System, CMZ, using editors called XEDIT and NOTEPAD. This all is running on the DECstation together with GEANT and several other applications.

In the CAD side, EUCLID is used as a test CAD system. EUCLID reads the SET files produced by the interface. EUCLID is running on VAX mainframe with the VMS operating system. All the files are transferred by FTP to VAX and back to the DECstation.

YSET SET file analyzer is used for syntax check, as well as to check violations of the SET standard.

5.3 Problems

Work with the GEANT-SET interface has not been without problems. There has been problems in different areas as follow:

- The computing Environment
- GEANT
- SET
- CAD

Some of those areas of problems have been in close connection with an other area, and had to be solved together with other problems.

5.3.1 The computing Environment

The problems with the computing Environment have mostly been involved with the hardware but there have been some software problems as well

In the beginning of the project we had problems with the disk space and the memory. The hard disk got full and we had to clean the hard disk from time to time. The memory was in the beginning insufficient and we have some problems with the swapping space and so on. The number of windows open was also limited because it increased the time of compiling. Later these problems have been solved and currently the DECstation is running fine.

Some software problems occurred when Digital Equipment Corporation was updating the operating system from ULTRIX 4.0 to 4.1. By accident we lost one of our back-up tapes and the DSPAW was down more than one week. We have had some problems with the networks and printers, but that seems to be usual in highly computerized environments.

5.3.2 GEANT

A delaying factor for the work with GEANT has been the lack of documentation. The documentation has been insufficient and out of date. Almost everything has been learned by asking and by experience. The updating of the documentation is under way and it will help the new collaborators to get started with GEANT. Another problem has been the heavy load on the key persons of GEANT.

GEANT has a concept of making replication of instances by dividing a mother volume according to certain parameters (See chapter 2). In a detector there is often a need for many instances of the same object. This enables very fast tracking. For us the division of a volume has been a major problem. There are still problems which have not been solved in the interface: How to handle a cone which is divided along an axis -which makes every result of the division different- and divided again. The problem is how to keep information of the real mother of every division.

We cannot extract the parameters of the result of the division from the GEANT data structure, since GEANT does not keep the parameters in the data structure, but calculates the result when needed. One way to handle this is the use of these existing GEANT subroutines to calculate the results for output. The other way is to implement the routines in the interface, but this requires a lot of code and time.

GEANT uses also negative parameters. When a parameter is negative, it means that the daughter volume fills the mother volume completely. The real value of this negative parameter can be found only by calculating.

The representation of reflected volume is also implemented in GEANT. The way how to represent it in SET is not yet implemented. The way to proceed will probably be to use some of the GEANT routines to calculate the parameters needed.

In GEANT, a user can position a volume inside its mother several times with different coordinates using a user routine called GSPOSP. The handling of these volumes is trivial, but is not implemented yet

in the GEANT-SET interface.

5.3.3 SET

SET standard is very large and is supposed to cover most ways to represent geometrical information. The exchange of that information between different CAD systems and GEANT seems to be almost impossible. The reason for that is that every CAD system, which has a SET interface, uses different subsets of the standard. Another problem is that very few CAD system have SET interfaces.

It is 'rather' easy to extract geometry from GEANT and convert it to SET, but the way back is quite difficult. For example when EUCLID reads data from SET file and after writes it back in SET, the result is completely different. The geometry is structured in a way that suits EUCLID, not GEANT.

SET file format is strictly sequential and it does not allow any forward references. A SET file has to read in a CAD system in its entirety and it is therefore not capable to transfer large amount of geometry information with the creation history into a CAD system. The SET file itself is able to transfer as much information as necessary, but EUCLID is not capable to handle all the information due to the limitation of memory.

5.3.4 CAD

EUCLID has been used as test CAD system for GEANT-SET interface. At the beginning everything was fine. When the sizes of the SET files became bigger, we faced a serious problem: the size of the data structure of EUCLID. EUCLID has limitations for points and blocks used, as well as the amount of Boolean operations.

GEANT is able to represent huge amount of volumes due to its way of doing replications in terms of division. This facility does not exist in the CAD world, and transfer of a full detector description is therefore necessary to get a complete model into the CAD system. although the SET file describing the model is relatively compact,

the memory used to represent the model inside EUCLID will easily exceed the current EUCLID data structure of 65000 points.

5.4 Solution

The GEANT-SET interface is writing the geometry of GEANT into SET file format in three basic phases:

- Write out the volume.
- Position the volume inside the mother volume.
- Write the tree (creation history) using Boolean operations.

In SET, all the volumes inside a volume must be first written out, then the volumes must be positioned inside the mother volume and after that the tree information can be written out in the form of Boolean operations.

5.4.1 The principle of the interface

GEANT starts to decode its geometrical data structure from the top of the tree. It starts from the mother volume and puts everything inside the mother. CAD systems start from the daughters and build the tree starting from the bottom. The interface had to be implemented so that it starts to decode the data structure of GEANT from the bottom of the GEANT tree structure. This is because the nature of SET.

The main program decodes the data structure of GEANT and when it finds a volume without contents, it writes the volume into SET file. When all of these volumes are written out, the program decodes the data structure again , now trying to find volumes, which contents are already written out. Then it positions the daughter volumes inside the mother. After that it writes out the mother volume. Now the tree structure can be written out and the programs

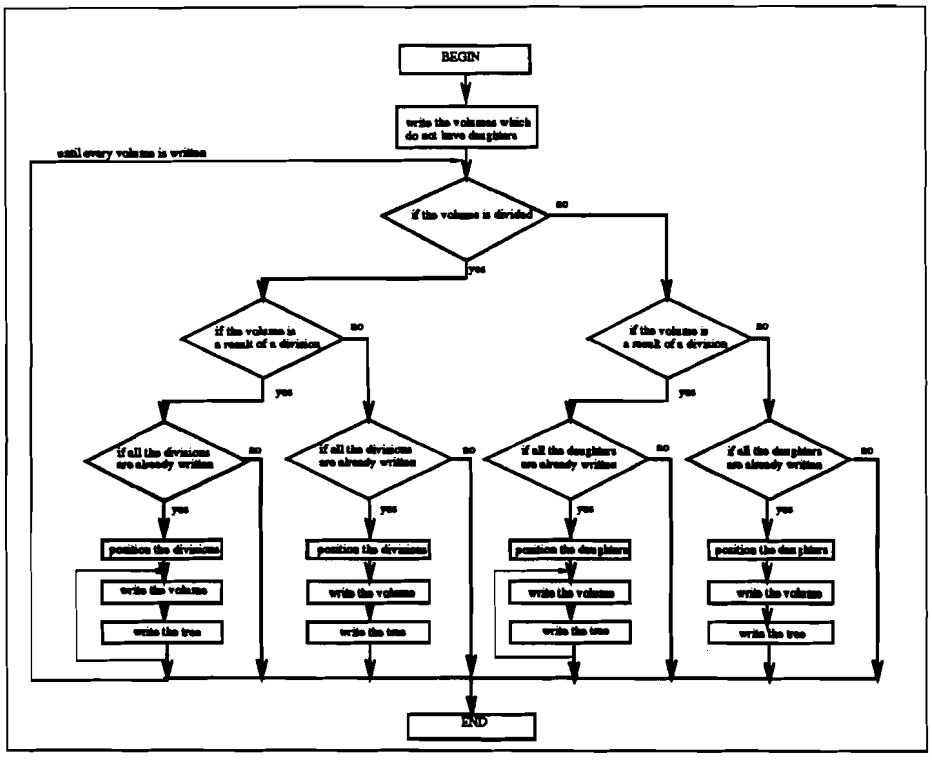


Figure 5.1: The simplified flow chart of the interface

writes the out the tree using Boolean operations. This cycle continuous until all the volumes are written out and the tree structure is complete.

The simplified flow chart is shown in figure 5.1 and the main program which decodes the data structure of GEANT can be found in appendix B-1.

5.4.2 Writing the volumes

Every different volume has its own subroutine for writing the volume into SET. So there is 16 different subroutines for writing the shape structure in SET. For an example, the simplest shape, a box, is written out in SET as follow:

```
@50,N1,:5,2#60,X,Y,Z
@302,N2#317,-DX,-DY,-DZ
@100,N3,:5,2,:9,'Name'#101,!N1,!N2
```

The meaning of that code is following:

- **@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 parallelpiped.
- **X,Y,Z** are the dimensions of the box.
- **@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 parallelpiped is first defined with the given parameters. Then the translation of coordinates is defined. The translation is necessary since GEANT and SET use origin in different place. At last the transformation operation is defined for a constructed solid which in this case is the translation of the coordinates of a primitive solid.

The whole subroutine which writes out the box can be find in appendix B-2 and the subroutine which writes out one of the most complicated basic shape, the polygon, can be find in appendix B-3.

5.4.3 Handling of the positioning and rotation

The positioning and rotation of a volume is handled by a subroutine, which writes out the rotation matrix and the position. The main program extracts the rotation matrix and the coordinates from the data structure and transfers those variables to the subroutine. The case that a volume is positioned several times inside the mother volume is not implemented yet.

The way the subroutine of the GEANT-SET interface writes the position and rotation of a volume is 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. This subroutine is in appendix B-4.

5.4.4 Tree decoding

Because of the nature of the SET file format, the tree structure in GEANT has to be decoded from bottom to top. It is not possible to carry the information of the **physical** tree structure in the memory, since the size of the intermediate data structure needed is too large. It was obligatory to write the tree information in SET continuously during the decoding of the data structure of GEANT.

The tree information is written in SET in terms of Boolean operations. The daughter volumes (contents) of the volume are first subtracted from the mother volume. Then the daughter volumes are positioned into the holes, which have been created by the subtraction. The tree information is written in SET, as follows:

```
@100,N1,:5,2#100,3,!MOTHER,!CHILD1,!CHILD2,...
@510,N2,:5,2#510,1,0,0,0,!N1,!CHILD1,!CHILD2,...
```

Where

- @100 is the block number which defines a constructed solid.
- @510 is the block number which defines a group of blocks defined by an associative relationship.
- N1,N2 are the sequence numbers of each block.

- `:5,2` is the dictionary entry number which gives a total subordination to one block.
- `!MOTHER` is the reference to sequence number of the block of the mother volume.
- `!CHILD` is the reference to sequence number of the block of the daughter volume.
- `#100` is the sub-block number for Boolean operation (3=subtraction).
- `#510` is the sub-block number for associative relationship.

In appendix B-5 is the code of the subroutine TREE.

5.4.5 Handling of the divisions

Divisions are used in GEANT because it corresponds to the way detectors are built and is to GEANT's advantage. A detector has several elements each appropriate to the particles which are expected in the region where the element will be placed. An element will be composed of cells, each corresponding to a data channel (or several data channels) and all the cells are identical, each containing, say, a converter plate, a sensewire, a vessel to contain the sensitive medium, the sensitive medium itself and so on. All of these cells are grouped together to form different subdetectors, i.e. the "external muon identifier", the "forward calorimeter", etc. From GEANT's point of view this is a simplification for it can quickly calculate in which division a particle is inside the volume and this tells in which datachannel the particle should show up - which is one of the things GEANT wants to know. There is a considerable saving of space and calculation for GEANT never needs to produce all the instances of the cells in any concrete way but merely transforms the coordinates of the particle in question to the coordinate system of the typical cell. For example the dataset describing the 1,200,000 volumes of L3 is represented in less than a megabyte.

Once the general shape of a detector element, say, the "forward calorimeter", is determined, this total space is divided according to the precision required in the location of particles, the efficiency of the sensitive volume, interaction length, cost etc.

There are three kinds of division to handle:

- The number of divisions has been given, so the axis is divided into that number of equal parts.
- The stepsize of the division and the number of divisions have been given, and divisions with that step are made from the beginning of the axis, leaving a remainder at the end.
- The stepsize and the number of divisions from a defined point on the axis have been given, leaving two remainders, one at each end of the axis.

Angular division is possible for axial symmetric objects about the axis. Radial division is possible where the boundaries are round the axis.

Each division may have daughters. A daughter may be a subdivision, and this subdivision may have a daughter which is a further subdivision. This process may continue as long as it is necessary to describe all the cells in the detector element.

However a daughter may also be a volume which is in each of the divisions, such as a converter plate, a sensewire etc. A daughter is only itemized once and is implied for each division. A volume can not be divided if it has positioned contents inside itself.

The handling of divisions is implemented in the GEANT-SET interface by checking if a volume is divided, if the mother volume of a volume is divided, and if the daughter volume of a volume is divided. All this cases must be checked separately.

The principle of writing the divisions in SET is the same as in the case of normal volumes. Writing the volumes, positioning volumes inside the mother volume, and the writing the tree information, is all same. The difference is, that the volume is now the result of a division, a slice, which is to be positioned inside the divided volume.

In case of a division, we use subroutines of GEANT to calculate the shape parameters and the coordinates of the result of a division. [5]

Chapter 6

Operating

6.1 General

The GEANT-SET interface is now in the phase that it is able to produce SET files from GEANT geometry with some exceptions. The interface can convert all the basic shapes of GEANT excluding one shape. It can also position volumes inside its mother volume and handle the rotation. The GEANT tree structure can be represented in SET file format. Handling of the divided volumes is also implemented in the interface, although with some exceptions.

The interface is only one-way interface. The reading of SET files into GEANT is not yet implemented, and that seems to be a more difficult task.

6.2 Testing

At the beginning of the development of the GEANT-SET interface, we used, as a first test model, a part from real detector, a sampling calorimeter. It is a box which contains other boxes and lots of divisions. The calorimeter drawn by GEANT without details is shown in figure 6.1.

Even if the shapes in the test model were very simple, it was too

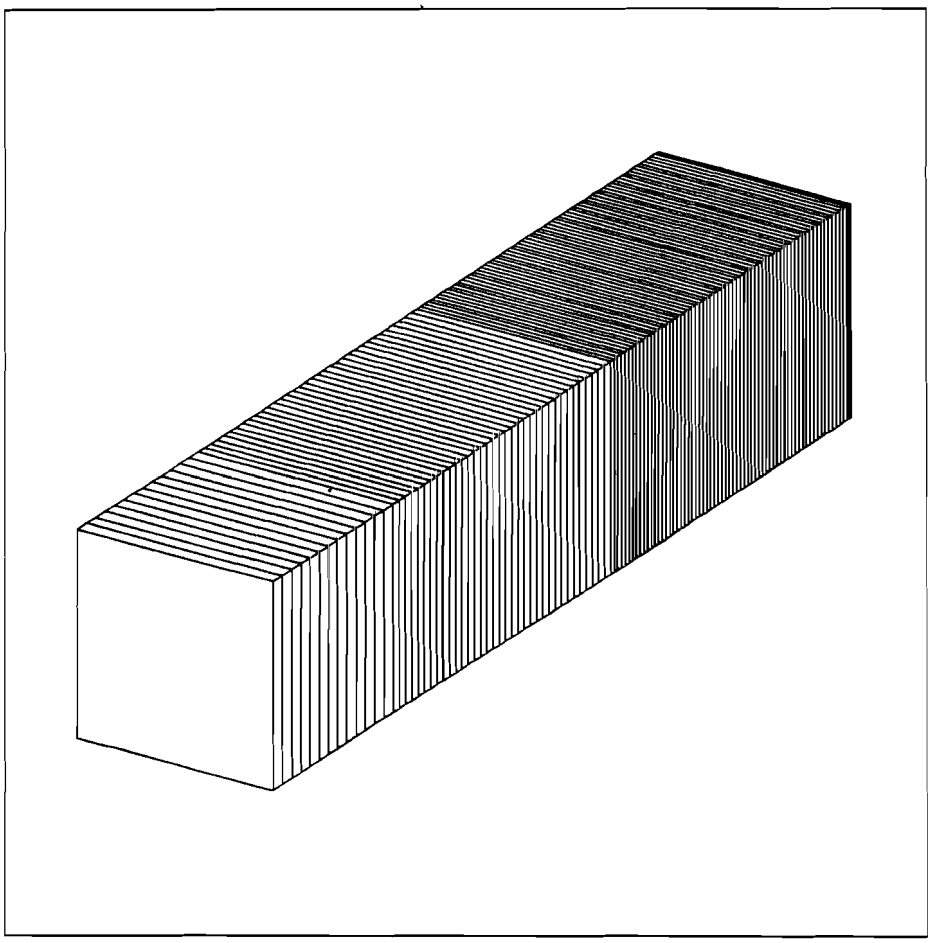


Figure 6.1: The calorimeter drawn by GEANT excluding lots of details

complicated for the first versions of the interface. We got some weird results and the SET file was quite big. It was difficult to find bugs from the interface, but the bugs were more than easy to see. Some typical test results at the beginning are shown in figures 6.2 and 6.2. Afterwards, when the interface was already working and tested, we faced the problem, that EUCLID is not capable to handle the amount of geometrical information that was in the calorimeter. The size of the data structure in GEANT which describes the geometry of the calorimeter was 7 kbytes and the size of the SET file was 4.2 Mbytes. This describes the data explosion when we write out all the divisions.

As a second try we started to use a very simple model. It was a box which contained some other boxes. The test geometry, 'Joke', is described in section 6.4.1.

In all test cases described below, we used first very simple geometry, then more complicated geometry and finally a part from a real detector.

6.2.1 Shapes

All the basic shapes of GEANT was tested separately. We created all basic shapes in GEANT, the interface wrote the shapes in SET and EUCLID read the SET files. All the basic shapes are shown in appendix A.

6.2.2 Positioning

For testing the interface and checking that it finds out positions and writes the correct tree information, we used first a simple box containing another box. Later we used more complicated geometry to find out the bugs. Finally we used part of a real detector, called DLS. The descriptions of DLS is written section 6.4.4.

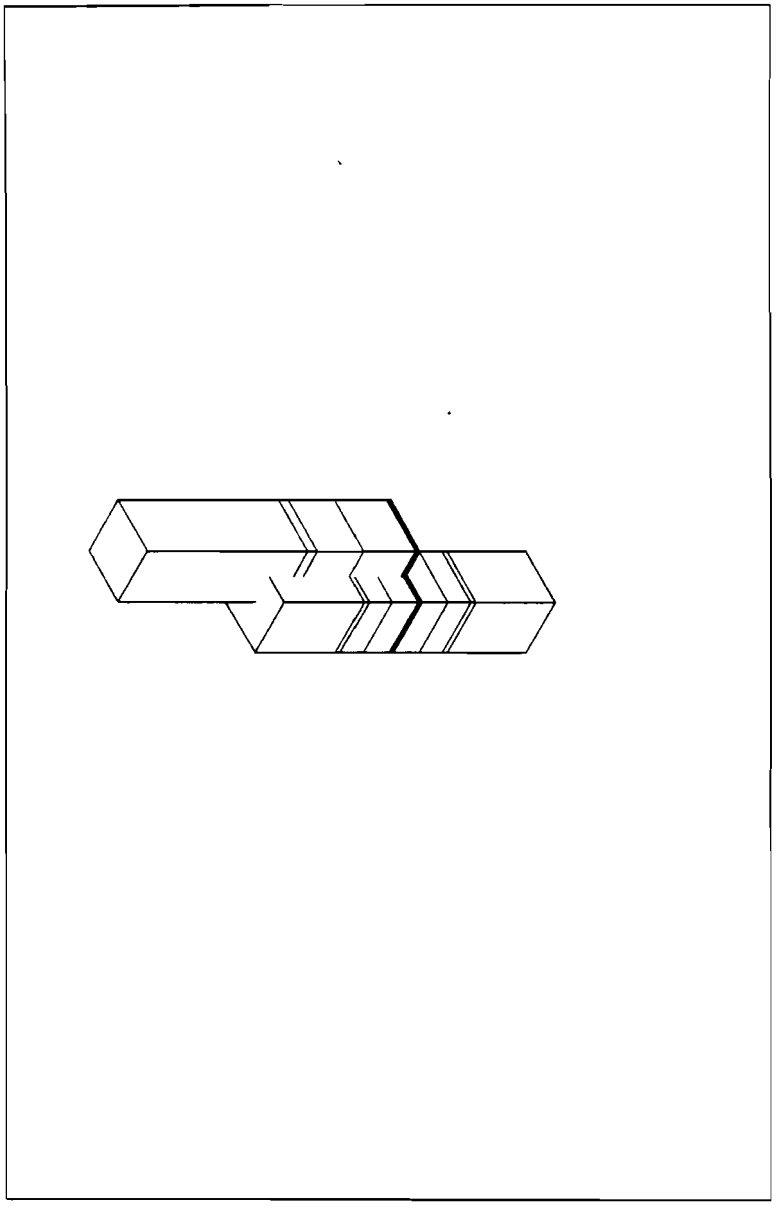


Figure 6.2: A test result of the calorimeter in EUCLID

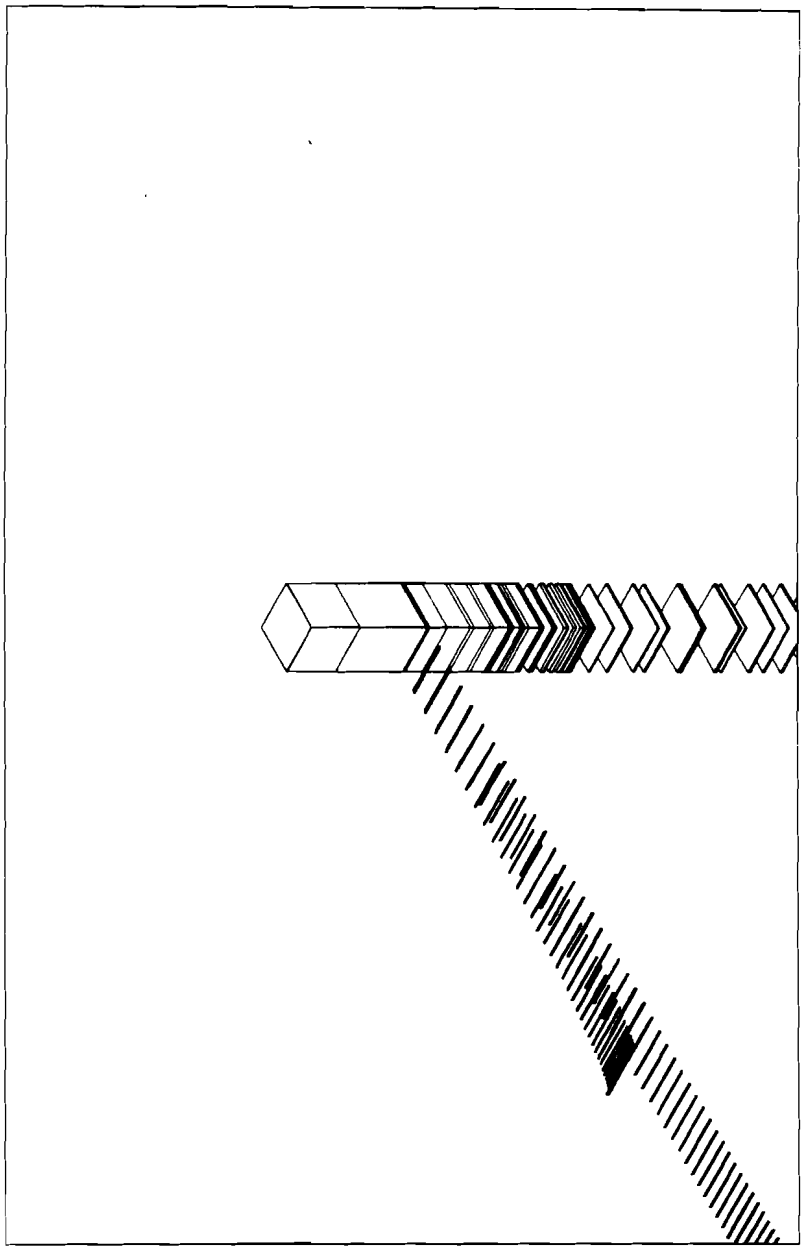


Figure 6.3: Another test result of the calorimeter in EUCLID

6.2.3 Divisions

In the case of divisions we had to start also from a very simple model. After we were able to use more complicated test geometry. We used also a model where a divided volume was divided further and further along different axis. Also in this case we finally used a part from a real detector.

6.3 Exceptions

The GEANT-SET interface is under development. In the life time of the interface the amount of exceptions is been reduced progressively. The further development will show if there is some exceptions which are impossible to solve.

In October 1991, the things currently not handled by the interface are following:

- **ELTU:** ELTU is one of the basic shapes in GEANT. It is converted into SET correctly, but because of a bug in EUCLID, we can not test or use it. EUCLID can not make a simple face with elliptical cross-section. This bug is known in MATRA DATAVISION.

- **GSPOSP:** The volumes positioned by GEANT user routine called GSPOSP, is not handled in the interface. With GSPOSP the user can position a volume inside its mother several times with different coordinates. The problem is trivial, and it will be soon implemented.

- **Divided CONE:** We not able to handle such divided volumes, where every division is different and further divided. (e.g. CONE, TRAP). This is because we are not able yet to carry information of each mother. This is basically depending of the size of the array needed. The solution for this has to be approached different way.

- **Reflection:** GEANT is able to represent a reflection of a volume and it is not yet been investigated, if there is in SET a simple way to solve that. In the worst case we have to do the calculations in the GEANT-SET interface.

6.4 Examples

I have chosen four different geometry to give an understanding of GEANT, SET and EUCLID. All models are described fully and only the descriptions of Nils and DLS are without the physical tree structure because of the limitation of memory in EUCLID.

6.4.1 Joke

The test model used in the beginning of the development of the GEANT-SET interface was a box containing one box in origin of the coordinate system of the mother. That box contained two equivalent boxes in opposite sides of their mother. Those boxes were created only once, but positioned twice. Both of those boxes contained a box in origin and again another box.

This test model, Joke, is shown in appendices as follow:

- **Appendix C-1:** Joke drawn by GEANT.
- **Appendix C-2:** Joke drawn by EUCLID.
- **Appendix C-3:** The logical tree structure of Joke in GEANT.
- **Appendix C-4:** The physical tree structure of Joke in EUCLID.
- **Appendix C-5:** The source code for creating Joke in GEANT.
- **Appendix C-6:** The SET file produced by the GEANT-SET interface of Joke.

6.4.2 Fede

The model, called Fede, is a box containing a trapezoid. The trapezoid contains a cylinder which is positioned twice. The cylinder contains a cone which further contains a sphere.

- **Appendix D-1:** Fede drawn by GEANT.
- **Appendix D-2:** Fede drawn by EUCLID.
- **Appendix D-3:** The logical tree structure of Fede in GEANT.
- **Appendix D-4:** The physical tree structure of Fede in EUCLID.
- **Appendix D-5:** The source code for creating Fede in GEANT.
- **Appendix D-6:** The SET file produced by the GEANT-SET interface of Fede.

6.4.3 Nils

Nils is a geometry, which was used to test the functionality of divisions. It is a box which contained a box and again another box positioned twice. This box was divided by three along the X axis. Every division was divided along the Y axis by four and again these divisions by five along the Z axis.

- **Appendix E-1:** Nils drawn by GEANT.
- **Appendix E-2:** Nils drawn by EUCLID.
- **Appendix E-3:** The logical tree structure of Nils in GEANT.
- **Appendix E-4:** The source code for creating Nils in GEANT.
- **Appendix E-5:** The SET file produced by the GEANT-SET interface of Nils.

6.4.4 DLS

DLS (Dilepton Spectrometer) is a part of a real detector. It was used in Lawrence Berkeley Laboratory experiment in Berkeley. The geometry is quite complicated to explain, but it can be seen from the following appendices:

- **Appendix F-1:** DLS drawn by GEANT.
- **Appendix F-2:** DLS drawn by EUCLID.
- **Appendix F-3:** The logical tree structure of DLS in GEANT.
- **Appendix F-4:** The source code for creating DLS in GEANT.
- **Appendix F-5:** The SET file produced by the GEANT-SET interface of DLS.

Chapter 7

Summary

The GEANT-SET interface is progressing continuously. The interface has been implemented in one direction, from GEANT to SET, although there might be some difficulties to implement some of the cases which are not yet handled. The way from SET to GEANT seems to be very difficult. We know now that it will be very difficult to transfer the whole detector geometry to a CAD system.

The problem that CAD systems are unable to handle the whole detector geometry, has to be solved in the near future. One of the possibilities is to implement different kind of modes in the interface. There can be a mode which transfers only a part of a detector which fits to CAD system. In fact, engineers are not interested in the whole detector, but only in one part of it at the time. That part of a detector might be chosen in GEANT from the tree structure, for example by clicking the mouse.

It is also possible to have a mode which transfers the whole detector geometry, but all the divisions are represented only once. The information of the divisions could be transferred by so called comment blocks of SET. This would reduce the amount of geometrical information enormously. This kind of mode in the interface is already implemented tentatively.

Another version of the interface, which we have already imple-

mented tentatively as well, is an interface, which writes out all volumes of a detector without the tree information. The SET file, produced by this kind of interface, was considerably larger and of course we lost all the creation history information.

One of the most interesting solutions came up recently. In that, it is proposed that GEANT is going to be developed more 'engineer friendly'. All the conceptual design could be done in GEANT. Some engineer work could be done in GEANT as well. If we are not able to come back from SET to GEANT, and we have to transfer all the geometrical information by hand, we must reduce the amount of geometrical information to be transferred. Engineers must be convinced of all the powerful features of GEANT to use it as a kind of CAD system, and to collaborate with physicists. After the design work in GEANT, engineers will have to use CAD systems only for manufacturing purposes etc.

Bibliography

[1] R. Brun, F. Bruyant, M. Maire, A.C. McPherson and P. Zanarini: *GEANT3* CERN DD/EE/84-1 (1986)

[2] French Standardization: *Data exchange and transfer standard specification (SET)*, Afnor (1989)

[3] Hiroaki Chiyokura: *Solid Modelling with designbase*, Ricoh Company Limited, Tokyo (1988)

[4] Nils-Joar Hoimyr: *CAD Interfaces in Research Enviroment*, The Norwegian Institute of Technology (1991)

[5] Mik Ferran: Reports from different CADD meetings and electronic mails, (1991)

[6] Michael E. Mortenson: *Geometric Modeling*, John Wiley & Sons, Inc (1985)

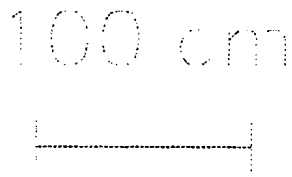
BOX

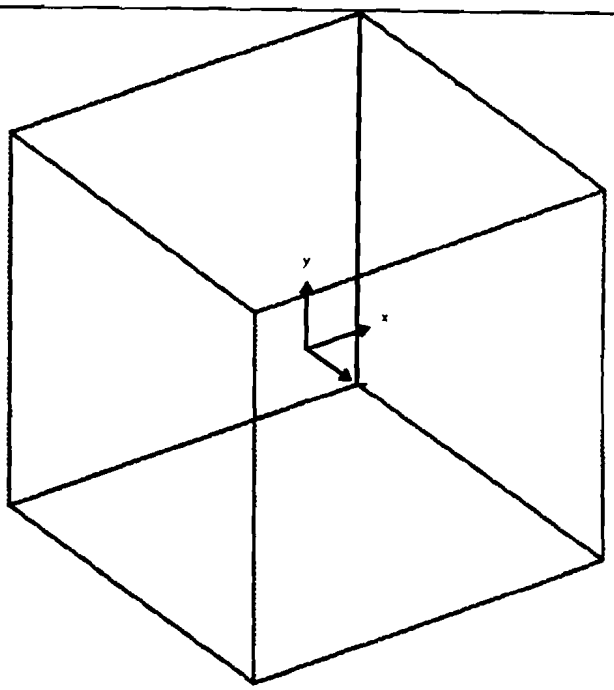
BOX specifications

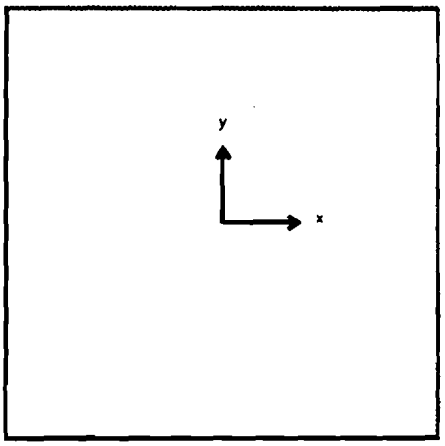
12/11/91

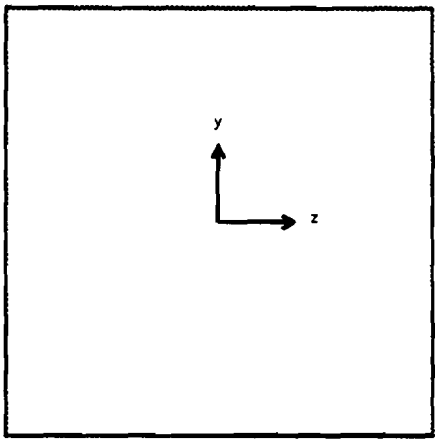
Box

DX = cm 100
DY = cm 100
DZ = cm 100









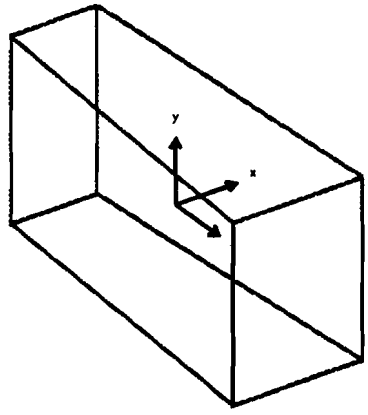
BOX

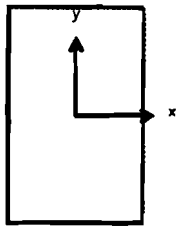
TRD1 specifications

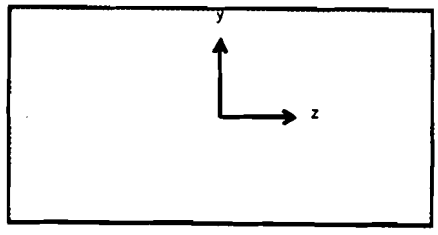
12/11/91

Trd1

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







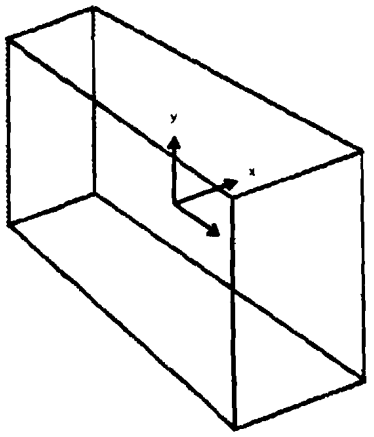
BOX

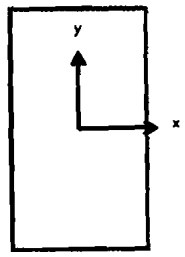
TRD2 specifications

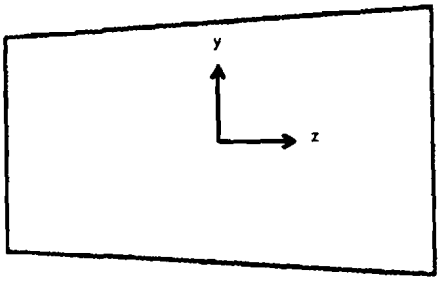
12/11/91

Trd2

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







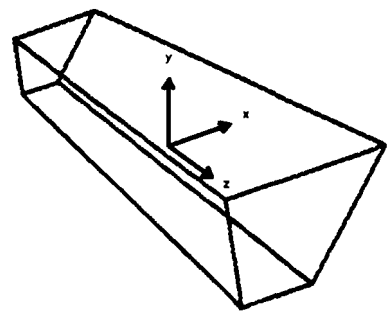
BOX

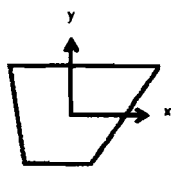
TRAP specifications

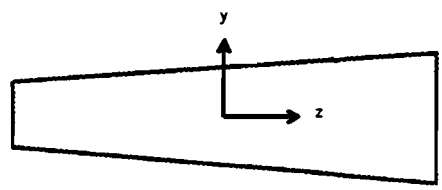
12/11/91

Trap

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







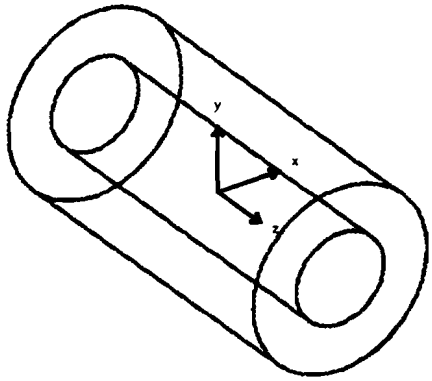
BOX

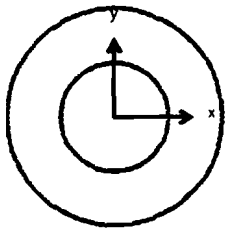
TUBE specifications

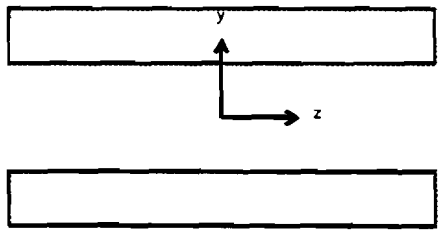
12/11/91

Tube

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







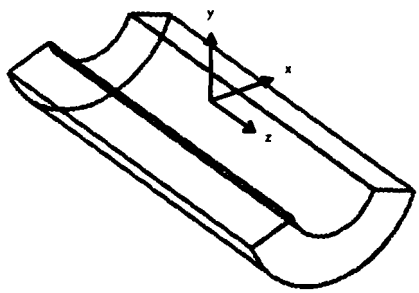
BOX

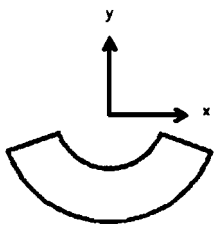
TUBS specifications

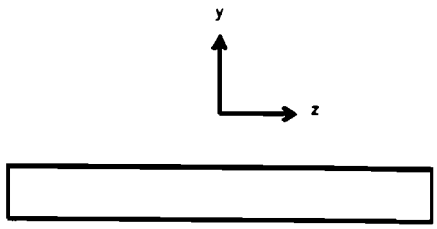
12/11/91

Tubs

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







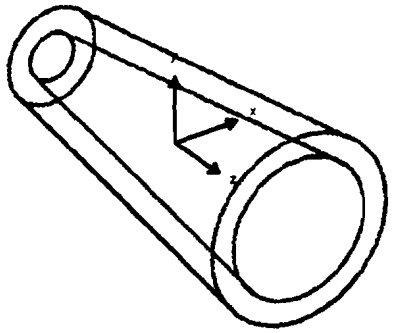
BOX

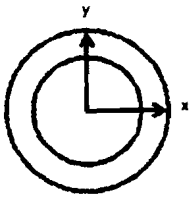
CONE specifications

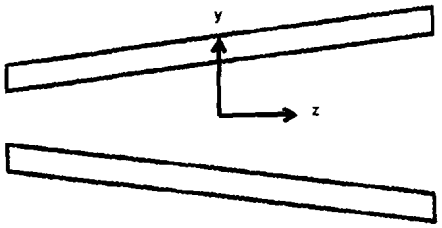
12/11/91

Cone

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







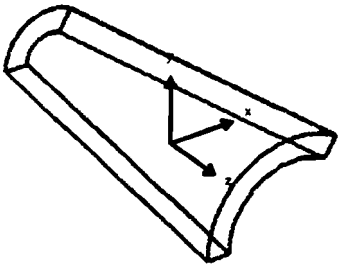
BOX

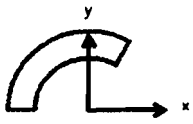
CONS specifications

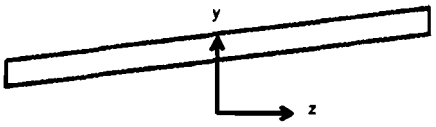
12/11/91

Cons

DZ	= cm	400
RMN1	= cm	50
RMX1	= cm	100
RMN2	= cm	150
RMX2	= cm	200
PHI1	= deg	60
PHI2	= deg	180







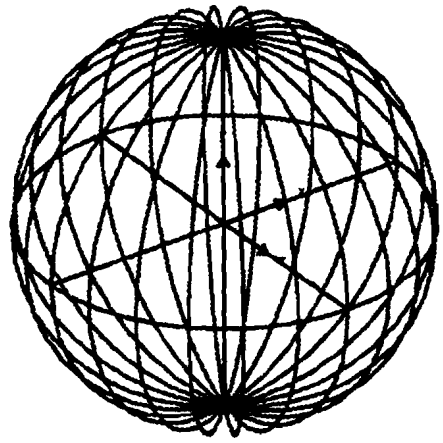
BOX

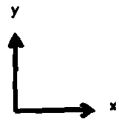
SPHE specifications

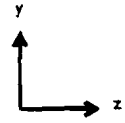
12/11/91

Sphe

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







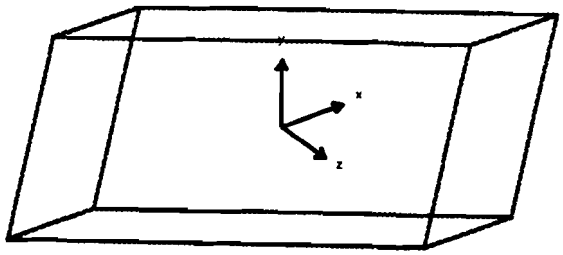
BOX

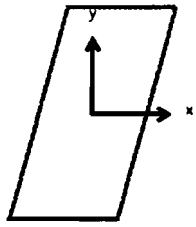
PARA specifications

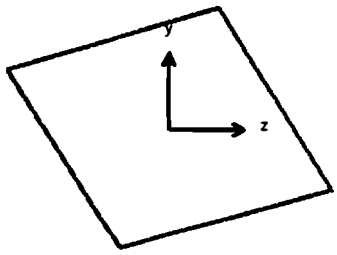
12/11/91

Para

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







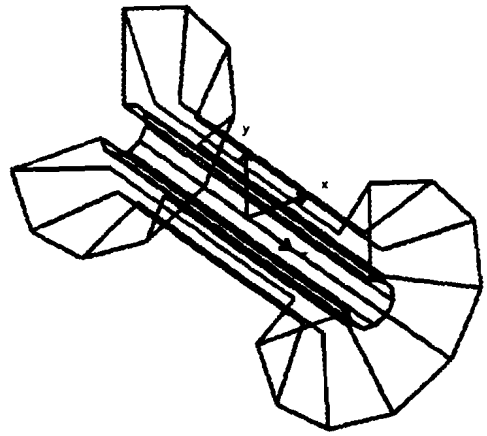
BOX

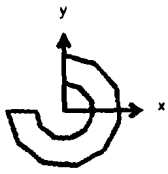
PGON specifications

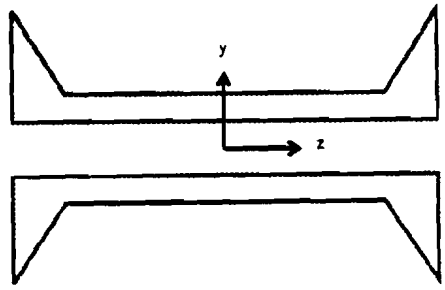
12/11/91

Pgon

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







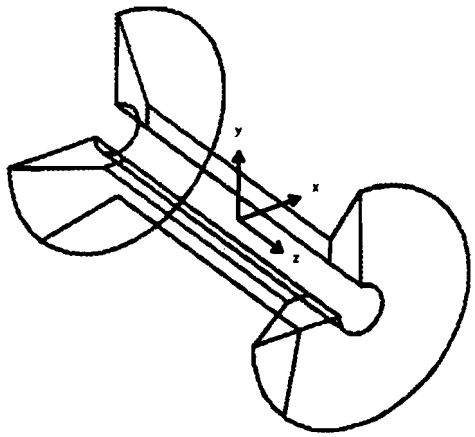
BOX

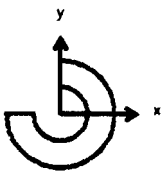
PCON specifications

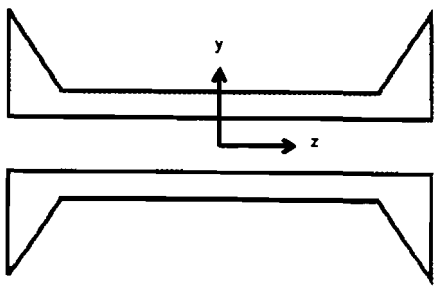
12/11/91

Pcon

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







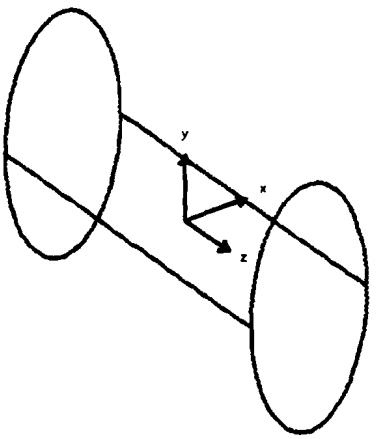
BOX

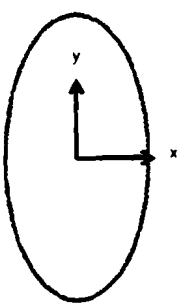
ELTU specifications

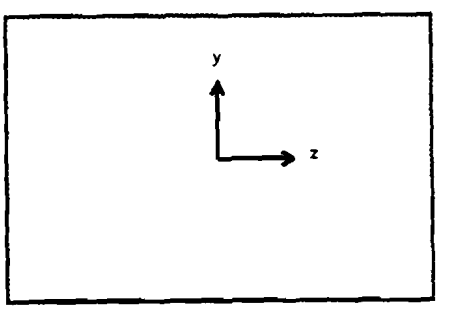
12/11/91

Eltu

A	= cm	100
B	= cm	200
DZ	= cm	300







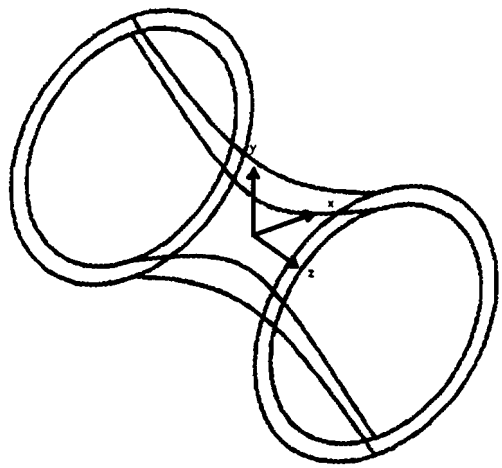
BOX

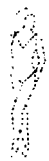
HYPE specifications

12/11/91

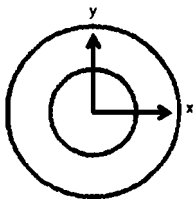
Hype

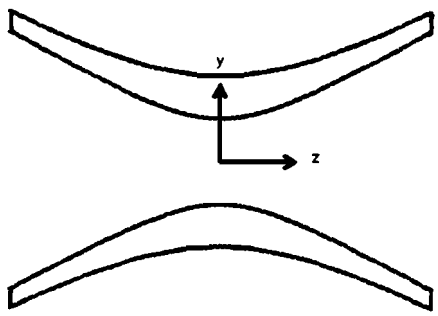
RMIN = cm	60
RMAX = cm	120
DZ = cm	300
TWST = deg	30





100 cm





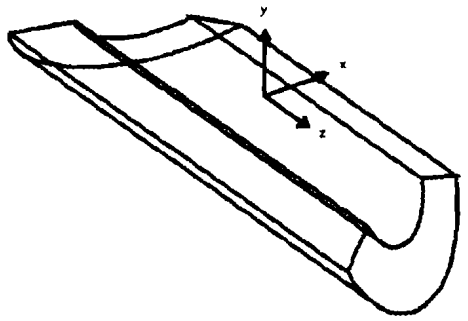
BOX

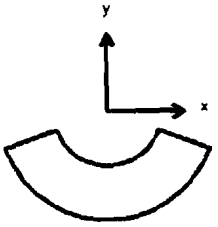
CTUB specifications

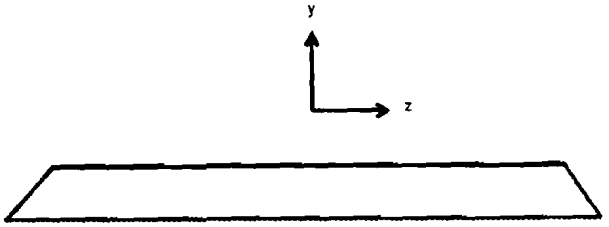
12/11/91

CTub

RMIN = cm	100
RMAX = cm	200
DZ = cm	400
PHI1 = deg	200
PHI2 = deg	340
LXL =	−150
LYL =	−200
LZL =	250
LXH =	150
LYH =	200
LZH =	320







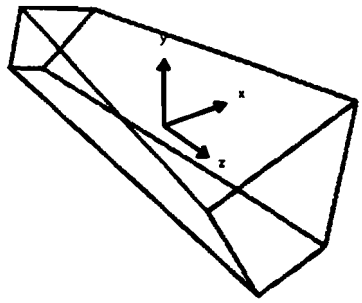
BOX

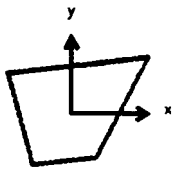
GTRA specifications

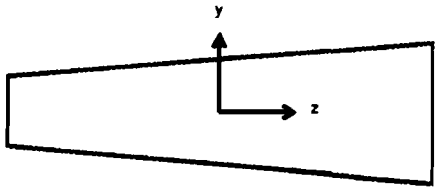
12/11/91

GTRA

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







```
*CMZ :          12/11/91  11.56.01  by  Jouko Vuoskoski
*-- Author :   NJ Hoimyr, J Vuoskoski
*
      SUBROUTINE GTOSET(F_NAME,A_NAME)
C.
C.    ******************************************************************
C.    *                                                                *
C.    *       Make the CG-Object with shape ISHAPE of parameters PAR   *
C.    *       with the same logic as GDRAWS                            *
C.    *       Converts GEANT SHAPES to SET                             *
C.    *                                                                *
C.    *                                                                *
C.    *                                                                *
C.    * SHAPE     SHAPE    SHAPE                                       *
C.    * NUMBER    TYPE     PARAMETERS                                  *
C.    * ---------------------------------------------------------------*
C.    *                                                                *
C.    *   1       BOX      DX,DY,DZ                                    *
C.    *   2       TRD1     DX1,DX2,DY,DZ                               *
C.    *   3       TRD2     DX1,DX2,DY1,DY2,DZ                          *
C.    *   4       TRAP     DZ,TX,TY,H1,BL1,TL1,TTH1,H2,BL2,TL2,TTH2    *
C.    *                                                                *
C.    *   5       TUBE     RMIN,RMAX,DZ                                *
C.    *   6       TUBS     RMIN,RMAX,DZ,PHIMIN,PHIMAX                  *
C.    *   7       CONE     DZ,RMIN1,RMAX1,RMIN2,RMAX2                  *
C.    *   8       CONS     DZ,RMIN1,RMAX1,RMIN2,RMAX2,PHIMIN,PHIMAX    *
C.    *                                                                *
C.    *   9       SPHE     RMIN,RMAX,THEMIN,THEMAX,PHIMIN,PHIMAX       *
C.    *                                                                *
C.    *  10       PARA     DX,DY,DZ,TXY,TXZ,TYZ                        *
C.    *  11       PGON     PHIMIN,DPHI,NDIV,NZ,Z(1),RMIN(1),RMAX(1),...*
C.    *  12       PCON     PHIMIN,DPHI,NZ,Z(1),RMIN(1),RMAX(1),Z(2),...*
C.    *                                                                *
C.    *       Authors : NJ Hoimyr, J Vuoskoski     *********          *
C.    *                                                                *
C.    ************************************* **************************
C.
*KEEP,GCBANK.
*KEEP,SETFIL.
*KEEP,SETCHR.
*KEND.
C
      CHARACTER*24  F_NAME
      CHARACTER*10  A_NAME
      CHARACTER*4   CHONLY,CGMATN,CMATNA
      INTEGER       NMP(100)
      INTEGER       NM(100,100)
      INTEGER       NCHILD(1000)
      DIMENSION     PAR(50)
      DIMENSION     PARMAT(10)
      DIMENSION     PARTRA(3) !for global coordinates
C-----------------------------------------------------------------
      WRITE(*,*)'GEANT-SET interface version 0.5 called'
      WRITE(*,*)'Starting to write SET file...'
      NVOL=0    ! number of volumes
      N1=2      ! counter for SET blocks
C
C       I n i t i a l i z e  S E T f i l e
C
      CALL START(F_NAME,A_NAME) ! Starting blocks
      NPLACE=1
C
C       c a l q.  n u m. o f  v o l u m e s
C
   10 IF(LQ(JVOLUM-NVOL-1).EQ.0.OR.IQ(JVOLUM-2).EQ.NVOL) GOTO 20
      NVOL=NVOL+1
```

```
      GOTO 10
   20 CONTINUE
C
C-----------------------------------------------------------
C-----------------------------------------------------------
C       first volumes without daughters (NIN2=0)
C       (bottom of the tree)
C
      DO 30 I=1, NVOL
        NM(I,1)=0        !clear seq-ivo  -variable
        NMP(I)=0         !clear 'already-positioned' -variable
   30 CONTINUE
C
C
       MATNAG=IQ(JVOLUM+1)     !Name of the mother of hole geometry
       CALL UHTOC(MATNAG,4,CGMATN,4)
C
C
      DO 40 LLVO=1, NVOL
        IVO1=LLVO
C
C          -----------information about the g-mother --------
C
        JVO1=LQ(JVOLUM-IVO1)       !Address of g-mother description
        NIN1=Q(JVO1+3)             !Number of mothers
C
C-----------------------------------------------------------
C          normal mothers
C
         IF (NIN1.GT.0) THEN
          DO 60 IIN1=1, NIN1
            JIN1=LQ(JVO1-IIN1)          !get adress mother description
            IVO2=Q(JIN1+2)              !get adress of mother volume
            JVO2=LQ(JVOLUM-IVO2)        !Address of mother description
            MATNA2=IQ(JVOLUM+IVO2)      !Name of the mother
            NIN2=Q(JVO2+3)              !Number of daughters
            ISHAP2=Q(JVO2+2)            !Shape number of the mother
C            NTRME2=Q(JVO2+4)           !Tracking media
C            NPARA2=Q(JVO2+5)           !Number of (shape) parameters
C
              IF (NIN2.EQ.0) THEN
                N1=N1+1             !Updates SET block sequence number
                CALL CALPAR(IVO2,NPAR,PAR)
                CALL SETSHP(ISHAP2,MATNA2,PAR)
                NM(IVO2,1)=N1
              ENDIF !(NIN2.EQ.0)
   60      CONTINUE
         ENDIF !(NIN1.GT.0)
C
C-------------------------------------------------------------
C          mothers which are results of a division
C
         IF (NIN1.LT.0) THEN  !G-mother divided volume and
                              !mother is a result of a division
           JDIV1=LQ(JVO1-1)   !address
           NDIV1=Q(JDIV1+3)   !number of division
           IVO2=Q(JDIV1+2)    !address of a result (mother)
C
           JVO2=LQ(JVOLUM-IVO2)       !Address of result (mother) description
           MATNA2=IQ(JVOLUM+IVO2)     !Name of the  result (mother)
           NIN2=Q(JVO2+3)             !Number of daughters
           ISHAP2=Q(JVO2+2)           !Shape number of the result (mother)
C
             IF (NIN2.EQ.0) THEN !no daughters
C
               CALL UHTOC(MATNA2,4,CMATNA,4)
```

```
              CALL GSATT('*','SEEN',0)
              CALL GSATT(CMATNA,'SEEN',-2)
              CALL SETDR4(CGMATN,MATNA2,NM,IVO2,NDIV1,ISHAP2)
C
C
   70         CONTINUE
            ENDIF
         ENDIF
   40 CONTINUE
C
C-----------------------------------------------------------
C-----------------------------------------------------------
C      then volumes which daughters already written
C
      DO 240 I=1, NVOL
C
      DO 140 LLVO=1, NVOL
      IVO1=LLVO
C
C          ----------information about the g-mother --------
C
C
        JVO1=LQ(JVOLUM-IVO1)       !Address of g-mother description
        NIN1=Q(JVO1+3)             !Number of mothers
C
C         ---------------- mother -------------------------
C          first normal mothers
C
         IF (NIN1.GT.0) THEN
          DO 160 IIN1=1, NIN1
            JIN1=LQ(JVO1-IIN1)         !get adress mother description
            IVO2=Q(JIN1+2)             !get adress of mother volume
            JVO2=LQ(JVOLUM-IVO2)       !Address of mother description
            MATNA2=IQ(JVOLUM+IVO2)     !Name of the mother
            NIN2=Q(JVO2+3)             !Number of daughters
            ISHAP2=Q(JVO2+2)           !Shape number of the mother
C            NTRME2=Q(JVO2+4)          !Tracking media
C            NPARA2=Q(JVO2+5)          ! Number of (shape) parameters
C
C-----------------------------------------------------------------------
C         check if the daughters already written
C         first normal daughters
C
            IF (NIN2.GT.0) THEN
              NINFLG=1
C
               DO 165 IIN2=1, NIN2
                JIN2=LQ(JVO2-IIN2)         !get adress daughter description
                IVO3=Q(JIN2+2)             !get adress of daughter volume
C
C         if not written yet or already posiotioned
C
                IF (NM(IVO3,1).EQ.0 .OR. NMP(IVO3).EQ.1) THEN
                 NINFLG=0
                ENDIF
  165         CONTINUE
C
C         if all the daughters already written, write also the
C         position, mother and tree
C
              IF (NINFLG.EQ.1) THEN
               DO 170 IIN2=1, NIN2
                JIN2=LQ(JVO2-IIN2)         !get adress daughter description
                IVO3=Q(JIN2+2)             !get adress of daughter volume
                JVO3=LQ(JVOLUM-IVO3)       !address of volume description
                MATNA3=IQ(JVOLUM+IVO3)     !get name
```

```
C                ISHAP3=Q(JVO3+2)          !Shape number
C                NTRME3=Q(JVO3+4)          !Tracking media
C                NPARA3=Q(JVO3+5)          !Number of (shape) parameters
C                NRCOP3=Q(JIN2+3)          !copy number
               MATRO3=Q(JIN2+4)            !matrix number
               XPOSIT=Q(JIN2+5)            ! \
               YPOSIT=Q(JIN2+6)            ! - p o s i t i o n
               ZPOSIT=Q(JIN2+7)            ! /
C
C          get parameters of rotation matrix
C
               CALL CALMAT(MATRO3,PARMAT)
C
C          write the position of the current daughter
C
               N1=N1+1        ! Updates SET block sequence number
               CALL POSIT(PARMAT,XPOSIT,YPOSIT,ZPOSIT,MATNA3,
     +NM(IVO3,1))
               NMP(IVO3)=1    ! now you can not pos. it again
C
C          create the seq. nb array for the tree
C
               NCHILD(IIN2)=N1
  170        CONTINUE
C
C        write the mother
C
             N1=N1+1
             CALL CALPAR(IVO2,NPAR,PAR)
             CALL SETSHP(ISHAP2,MATNA2,PAR)
             NM(IVO2,1)=N1
C
C        create the tree
C
             N1=N1+1              ! Updates SET block sequence number
             CALL TREE(NCHILD,NIN2,NM(IVO2,1),MATNA2,IVO2)
             NM(IVO2,1)=N1      ! this is now the new volume
C
            ENDIF  !(NINFLG.EQ.1)
          ENDIF  !(NIN2.GT.0)
C
C-----------------------------------------------------------------------------79
C        check if the daughters already written
C        then daughters which are results of a division
C        (mother is divided)
C
          IF (NIN2.LT.0) THEN
            NINFLG=1
C
            JDIV2=LQ(JVO2-1)      !address
            NDIV2=Q(JDIV2+3)      !number of division
            IVO3=Q(JDIV2+2)       !adress of daughter volume
            JVO3=LQ(JVOLUM-IVO3) !address of result (daughter) description
            MATNA3=IQ(JVOLUM+IVO3)     ! get name
C
C        if not written yet or already posiotioned
C
               IF (NM(IVO3,1).EQ.0 .OR. NMP(IVO3).EQ.1) THEN
                NINFLG=0
               ENDIF
C
C        if all the daughters already written, write also the
C        position, mother and tree
C
            IF (NINFLG.EQ.1) THEN
C
```

```
C           first global coordinates for mother
C
               CALL UHTOC(MATNA2,4,CMATNA,4)
               CALL GSATT('*','SEEN',0)
               CALL GSATT(CMATNA,'SEEN',-2)
               CALL SETDR2(CGMATN,PARTRA)
C
C           then position every division
C
               CALL UHTOC(MATNA3,4,CMATNA,4)
               CALL GSATT('*','SEEN',0)
               CALL GSATT(CMATNA,'SEEN',-2)
               CALL SETDRW(CGMATN,MATNA3,NM,NCHILD,IVO3,NDIV2,PARTRA)
               NMP(IVO3)=1    ! now you can not pos. it again
C
C
C         write the mother
C
               N1=N1+1
               CALL CALPAR(IVO2,NPAR,PAR)
               CALL SETSHP(ISHAP2,MATNA2,PAR)
               NM(IVO2,1)=N1
C
C         create the tree
C
               N1=N1+1             ! Updates SET block sequence number
               CALL TREE(NCHILD,NDIV2,NM(IVO2,1),MATNA2,IVO2)
               NM(IVO2,1)=N1       ! this is now the new volume
C
             ENDIF  !(NINFLG.EQ.1)
           ENDIF  !(NIN2.LT.0)
  160      CONTINUE
       ENDIF  !(NIN1.GT.0)
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C          then mothers which are results of a division
C          (g-mother is divided)
C
         IF (NIN1.LT.0) THEN
C
           JDIV1=LQ(JVO1-1)           !address
           NDIV1=Q(JDIV1+3)           !number of division
           IVO2=Q(JDIV1+2)            !address of a result (mother)
           JVO2=LQ(JVOLUM-IVO2)       !Address of result (mother) description
           MATNA2=IQ(JVOLUM+IVO2)     !Name of the  result (mother)
           NIN2=Q(JVO2+3)             !Number of daughters
           ISHAP2=Q(JVO2+2)           !Shape number of the result (mother)
C           NTRME2=Q(JVO2+4)          !Tracking media
C           NPARA2=Q(JVO2+5)          !Number of (shape) parameters
C
C-----------------------------------------------------------------------
C         check if the daughters already written
C         first normal daughters
C
           IF (NIN2.GT.0) THEN
             NINFLG=1
C
               DO 260 IIN2=1, NIN2
                JIN2=LQ(JVO2-IIN2)          ! get adress daughter description
                IVO3=Q(JIN2+2)              ! get adress of daughter volume
C
C         if not written yet or already posiotioned
C
                 IF (NM(IVO3,1).EQ.0 .OR. NMP(IVO3).EQ.1) THEN
                  NINFLG=0
```

```
                ENDIF
  260         CONTINUE
C
C         if all the daughters already written, write also the
C         position, mother and tree
C
              IF (NINFLG.EQ.1) THEN
               DO 370 IIN2=1, NIN2
                JIN2=LQ(JVO2-IIN2)          !get adress daughter description
                IVO3=Q(JIN2+2)              !get adress of daughter volume
                JVO3=LQ(JVOLUM-IVO3)        !address of volume description
                MATNA3=IQ(JVOLUM+IVO3)      !get name
C                ISHAP3=Q(JVO3+2)           !Shape number
C                NTRME3=Q(JVO3+4)           !Tracking media
C                NPARA3=Q(JVO3+5)           !Number of (shape) parameters
C                NRCOPY=Q(JIN2+3)           !copy number
                MATRO3=Q(JIN2+4)            !matrix number
                XPOSIT=Q(JIN2+5)            ! \
                YPOSIT=Q(JIN2+6)            ! - p o s i t i o n
                ZPOSIT=Q(JIN2+7)            ! /
C
C           get parameters of rotation matrix
C
                CALL CALMAT(MATRO3,PARMAT)
C
C           write the position of the current daughter
C
                N1=N1+1        ! Updates SET block sequence number
                CALL POSIT(PARMAT,XPOSIT,YPOSIT,ZPOSIT,MATNA3,
     +NM(IVO3,1))
                NMP(IVO3)=1    ! now you can not pos. it again
C
C           create the seq. nb array for the tree
C
                NCHILD(IIN2)=N1
  370          CONTINUE
C
C         write the mothers (results of a division)
C         and create the tree
C
               CALL UHTOC(MATNA2,4,CMATNA,4)
               CALL GSATT('*','SEEN',0)
               CALL GSATT(CMATNA,'SEEN',-2)
               CALL SETDR3(CGMATN,MATNA2,NM,NCHILD,IVO2,NDIV1,
     +NIN2,ISHAP2)
C
C
              ENDIF  !(NINFLG.EQ.1)
            ENDIF  !(NIN2IDIV.GT.0)
C
C-----------------------------------------------------------------------------79
C         check if the daughters already written
C         then daughters which are results of a division
C         (mother is divided and also a result of a division)
C
            IF (NIN2.LT.0) THEN
              NINFLG=1
C
              JDIV2=LQ(JVO2-1)    !address
              NDIV2=Q(JDIV2+3)    !number of division
              IVO3=Q(JDIV2+2)     !
              JVO3=LQ(JVOLUM-IVO3) !address of result (daughter) description
              MATNA3=IQ(JVOLUM+IVO3)     ! get name
C
C         if not written yet or already posiotioned
C
```

```
                IF (NM(IVO3,1).EQ.0 .OR. NMP(IVO3).EQ.1) THEN
                 NINFLG=0
                ENDIF
C
C         if all the daughters already written, write also the
C         position, mother and tree
C
              IF (NINFLG.EQ.1) THEN
C
C           first global coordinates for mother
C
                CALL UHTOC(MATNA2,4,CMATNA,4)
                CALL GSATT('*','SEEN',0)
                CALL GSATT(CMATNA,'SEEN',-2)
                CALL SETDR2(CGMATN,PARTRA)
C
C           then position every division
C
                CALL UHTOC(MATNA3,4,CMATNA,4)
                CALL GSATT('*','SEEN',0)
                CALL GSATT(CMATNA,'SEEN',-2)
                CALL SETDRW(CGMATN,MATNA3,NM,NCHILD,IVO3,NDIV2,PARTRA)
                NMP(IVO3)=1    ! now you can not pos. it again
C
C
C         write all the mothers (results of a division)
C         and create the tree
C
C
                CALL UHTOC(MATNA2,4,CMATNA,4)
                CALL GSATT('*','SEEN',0)
                CALL GSATT(CMATNA,'SEEN',-2)
                CALL SETDR3(CGMATN,MATNA2,NM,NCHILD,IVO2,NDIV1,
     +NDIV2,ISHAP2)
C
              ENDIF  !(NINFLG.EQ.1)
            ENDIF  !(NIN2.LT.0)
C
        ENDIF  !(NIN1.LT.0)
C
C
  140 CONTINUE
  240 CONTINUE
C
C-----------------------------------------------------------------
C-----------------------------------------------------------------
C          finally the g-mother
C
            JVO1=LQ(JVOLUM-1)          !Address of mother description
            MATNA1=IQ(JVOLUM+1)        !Name of the mother
            NIN1=Q(JVO1+3)             !Number of daughters
            ISHAP1=Q(JVO1+2)           !Shape number of the mother
C            NTRME1=Q(JVO1+4)          !Tracking media
C            NPARA1=Q(JVO1+5)          !Number of (shape) parameters
C
C
            IF (NIN1.EQ.0) THEN !   only one volume in whole geometry
C
C         write the VOLUME
                N1=N1+1
                CALL CALPAR(1,NPAR,PAR)
                CALL SETSHP(ISHAP1,MATNA1,PAR)
                NM(1,1)=N1
           ENDIF
C
C-----------------------------------------------------------------
```

```
C         first normal mothers
C
            IF (NIN1.GT.0) THEN
C
               DO 770 IIN1=1, NIN1
                JIN1=LQ(JVO1-IIN1)        !get adress daughter description
                IVO2=Q(JIN1+2)            !get adress of daughter volume
                JVO2=LQ(JVOLUM-IVO2)      !address of volume description
                MATNA2=IQ(JVOLUM+IVO2)    !get name
C                ISHAP2=Q(JVO2+2)         !Shape number
C                NTRME2=Q(JVO2+4)         !Tracking media
C                NPARA2=Q(JVO2+5)         !Number of (shape) parameters
C                NRCOPY=Q(JIN1+3)         !copy number
                MATRO2=Q(JIN1+4)          !matrix number
                XPOSIT=Q(JIN1+5)          ! \
                YPOSIT=Q(JIN1+6)          ! - p o s i t i o n
                ZPOSIT=Q(JIN1+7)          ! /
C
C           get parameters of rotation matrix
C
                CALL CALMAT(MATRO2,PARMAT)
C
C           write the position of the current mother
C
                N1=N1+1        ! Updates SET block sequence number
                CALL POSIT(PARMAT,XPOSIT,YPOSIT,ZPOSIT,MATNA2,
     +NM(IVO2,1))
C
C
                NCHILD(IIN1)=N1
  770          CONTINUE
C
C         write the g-mother
C
              N1=N1+1
              CALL CALPAR(1,NPAR,PAR)
              CALL SETSHP(ISHAP1,MATNA1,PAR)
              NM(1,1)=N1
C
C         create the tree
C
              N1=N1+1             ! Updates SET block sequence number
              CALL TREE(NCHILD,NIN1,NM(1,1),MATNA1,1)
C
            ENDIF  !(NIN1.GT.0)
C
C
C-----------------------------------------------------------------------------79
C         then mothers which are results of a division
C         (g-mother is divided)
C
            IF (NIN.LT.0) THEN
C
              JDIV1=LQ(JVO1-1 )     !address
              NDIV1=Q(JDIV1+3)      !number of division
              IVO2=Q(JDIV1+2)       !
              JVO2=LQ(JVOLUM-IVO2) !address of result (daughter) description
              MATNA2=IQ(JVOLUM+IVO2)     ! get name
C
C
C           first global coordinates for mother
C
              CALL UHTOC(MATNA1,4,CMATNA,4)
              CALL GSATT('*','SEEN',0)
              CALL GSATT(CMATNA,'SEEN',-2)
              CALL SETDR2(CGMATN,PARTRA)
```

```
C
C           then position every division
C
               CALL UHTOC(MATNA2,4,CMATNA,4)
               CALL GSATT('*','SEEN',0)
               CALL GSATT(CMATNA,'SEEN',-2)
               CALL SETDRW(CGMATN,MATNA2,NM,NCHILD,IVO2,NDIV1,PARTRA)
C
C         write the g-mother
C
               N1=N1+1
               CALL CALPAR(1,NPAR,PAR)
               CALL SETSHP(ISHAP1,MATNA1,PAR)
               NM(1,1)=N1
C
C         create the tree
C
               N1=N1+1             ! Updates SET block sequence number
               CALL TREE(NCHILD,NDIV1,NM(1,1),MATNA1,1)
C
        ENDIF  !(NIN1.LT.0)
C-------------------------------------------------------------

C
C
C
C    E n d  o f  S E T f i l e
C
      CALL ENDSET
C
      WRITE(*,*)'The detector geometry is now converted into SET'
      WRITE(*,*)'format and the file is in the working directory'
C
      END
```

```
*CMZ :          06/11/91  14.35.10  by  Jouko Vuoskoski
*-- Author :  Nils Joar Hoimyr, Jouko Vuoskoski
C
      SUBROUTINE BOXSET(MATNAM,PAR,ISHAPE)
C     Created  12-JUN-1991   Nils Joar Hoimyr, Jouko Vuoskoski
C
C  Takes the GEANT shape BOX, converts it to a SET box, with
C  the right position and writes the shape to the current SET file.
C
C-----------------------------------------------------------------------
C
*KEEP,SETFIL.
      COMMON/SETFIL/NPLACE,N_UNIT,N1
C
*KEEP,SETCHR.
      COMMON/GEOMTC/BLKSTR  ! this must put in every routine
      CHARACTER*500 BLKSTR   ! no use to be common variable
C
*KEND.
C
      DIMENSION PAR(50)
C
      REAL*8 X,Y,Z
C
C
C-----------------------------------------------------------------------
C
C             BOX
C
         DX=PAR(1)
         DY=PAR(2)
         DZ=PAR(3)
         X=2*DX
         Y=2*DY
         Z=2*DZ
C *WRITE SET @50,N1,:5,2#60,X,Y,Z
C *WRITE SET @302,N2,#317,-DX,-DY,-DZ
C *WRITE SET @100,N3,:5,2,:9,'Name'#101,!N1,!N2
C
C-----------------------------------------------------------------------
C      this is for writing without spaces and line length of 80
C
        DO 500 I=1, 500
          BLKSTR(I:I)=' '
  500   CONTINUE
        J=NPLACE
        WRITE(BLKSTR,10010)N1, X, Y, Z
        DO 600 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
  600   CONTINUE
        N1=N1+1
        DO 700 I=1, 500
          BLKSTR(I:I)=' '
  700   CONTINUE
        WRITE(BLKSTR,10020)N1,-DX,-DY,-DZ
        DO 800 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
```

```
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
  800     CONTINUE
          DO 900 I=1, 500
            BLKSTR(I:I)=' '
  900     CONTINUE
          N1=N1+1
          WRITE(BLKSTR,10030)N1,MATNAM,N1-2,N1-1
          DO 1000 I=1, 500
            IF (BLKSTR(I:I) .NE. ' ') THEN
              WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
                IF (J .EQ. 80) THEN
                  WRITE(N_UNIT,'(/,$)')
                  J=0
                END IF
              J=J+1
            END IF
 1000     CONTINUE
10010     FORMAT('@50,',I,',:5,2#60,',F,',',F,',',F)
10020     FORMAT('@302,',I,'#317,',F,',',F,',',F)
10030     FORMAT('@100,',I,',:5,2,:9,''',A4,'''#101,!',I,',!',I)
C
          NPLACE=J
C
C
        END
```

```
*CMZ :          06/11/91  12.15.16  by  Jouko Vuoskoski
*-- Author : Nils Joar Hoimyr, Jouko Vuoskoski
C
      SUBROUTINE PGOSET(MATNAM,PAR,ISHAPE)
C
C     Created  26-JUN-1991   Nils Joar Hoimyr, Jouko Vuoskoski
C-----------------------------------------------------------
C  Calculates the sectional face of 1 cell of the PGON shape.   This face is
C  copied and rotated around the Z-axis to make "the other side" of the first
C  cell.  A ruled solid is generated between the 2 faces, to created the cell.
C  The rest of the cells are created by copying and rotation of the first cell.
C  The final result is obtained with a Boolean fusion of the cells.  (The whole
C  creation history is written to the SET file.)
C-----------------------------------------------------------
C
*KEEP,SETFIL.
      COMMON/SETFIL/NPLACE,N_UNIT,N1
C
*KEEP,SETCHR.
      COMMON/GEOMTC/BLKSTR  ! this must put in every routine
      CHARACTER*500 BLKSTR   ! no use to be common variable
C
*KEND.
      DIMENSION PAR(50)
      REAL*4 PX,PY,PZ,PHIC,PHI1
      REAL*4 R1,R2,R3,R4,R5,R6,R7,R8,R9
C
C-----------------------------------------------------------------
C
C        Calculates range of each cell
C
         PHI1=  (PAR(1)+180)/180*3.14159265359
         NZ=    PAR(4)
         NPDV= PAR(3)
         PHIC= (PAR(2)/PAR(3))/180*3.14159265359
C       Rotation around the Z-axis.  Coeffisients of rotation:
         R1=  COS(PHI1)
         R2= -SIN(PHI1)
         R3=  0.0
         R4=  SIN(PHI1)
         R5=  COS(PHI1)
         R6=  0.0
         R7=  0.0
         R8=  0.0
         R9=  1.0
C       Face defined in the yz-plane (x=0)
C-----------------------------------------------------------------
C       This is for writing without spaces and line length of 80
         DO 400 I=1, 500
           BLKSTR(I:I)=' '
  400    CONTINUE
         J=NPLACE
         WRITE(BLKSTR,10005)N1
         DO 450 I=1, 500
           IF (BLKSTR(I:I) .NE. ' ') THEN
             WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
               IF (J .EQ. 80) THEN
                 WRITE(N_UNIT,'(/,$)')
                 J=0
               END IF
             J=J+1
           END IF
  450    CONTINUE
C-----------------------------------------------------------
         N3= 1
         PX= 0.0
```

```
C
         PY= PAR(6)
         PZ= PAR(5)
C-----------------------------------------------------------
C       This is for writing without spaces and line length of 80
         DO 500 I=1, 500
           BLKSTR(I:I)=' '
  500    CONTINUE
         WRITE(BLKSTR,10010)PX,PY,PZ
         DO 600 I=1, 500
           IF (BLKSTR(I:I) .NE. ' ') THEN
             WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
               IF (J .EQ. 80) THEN
                 WRITE(N_UNIT,'(/,$)')
                 J=0
               END IF
             J=J+1
           END IF
  600    CONTINUE
C-----------------------------------------------------------
C
         DO 100 K=1,NZ
          N3=N3+3
          PY= PAR(N3+2)
          PZ= PAR(N3)
C
C-----------------------------------------------------------
C       This is for writing without spaces and line length of 80
         DO 720 I=1, 500
           BLKSTR(I:I)=' '
  720    CONTINUE
         WRITE(BLKSTR,10010)PX,PY,PZ
         DO 810 I=1, 500
           IF (BLKSTR(I:I) .NE. ' ') THEN
             WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
               IF (J .EQ. 80) THEN
                 WRITE(N_UNIT,'(/,$)')
                 J=0
               END IF
             J=J+1
           END IF
  810    CONTINUE
C-----------------------------------------------------------
C
  100    CONTINUE
         PY= PAR(N3+1)
         PZ= PAR(N3)
C
C-----------------------------------------------------------
C       This is for writing without spaces and line length of 80
         DO 840 I=1, 500
           BLKSTR(I:I)=' '
  840    CONTINUE
         WRITE(BLKSTR,10010)PX,PY,PZ
         DO 860 I=1, 500
           IF (BLKSTR(I:I) .NE. ' ') THEN
             WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
               IF (J .EQ. 80) THEN
                 WRITE(N_UNIT,'(/,$)')
                 J=0
               END IF
             J=J+1
           END IF
  860    CONTINUE
C
C       Geometric Transformation
```

```
C*        WRITE SET @302,N1..#301,R1,R2,R3,R4,R5,R6,R7,R8,R9
C
       N1=N1+1
       NG= N1
        DO 700 I=1, 500
          BLKSTR(I:I)=' '
  700   CONTINUE
        WRITE(BLKSTR,10020)N1,R1,R2,R3,R4,R5,R6,R7,R8,R9
        DO 800 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
  800   CONTINUE
C---First FACE:
C*        WRITE SET @113,F2..#101,!F1,N1
C-----------------------------------------------
       DO 820 I=1, 500
          BLKSTR(I:I)=' '
  820   CONTINUE
        N1=N1+1
        WRITE(BLKSTR,10030)N1,N1-2,N1-1
        DO 830 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
 830   CONTINUE
C-------------------------------------------------------------
C      Next step is to obtain the right position of the second face to
C      create a cell:
C      Rotation around the Z-axis.  Coeffisients of rotation:
         R1=  COS(PHIC)
         R2= -SIN(PHIC)
         R3=  0.0
         R4=  SIN(PHIC)
         R5=  COS(PHIC)
         R6=  0.0
         R7=  0.0
         R8=  0.0
         R9=  1.0
C
C      Geometric Transformation
C*        WRITE SET @302,N1..#301,R1,R2,R3,R4,R5,R6,R7,R8,R9
C
       N1=N1+1
       NG= N1
        DO 880 I=1, 500
          BLKSTR(I:I)=' '
  880   CONTINUE
        WRITE(BLKSTR,10020)N1,R1,R2,R3,R4,R5,R6,R7,R8,R9
        DO 890 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
```

```
            J=J+1
          END IF
  890   CONTINUE
C
C       Second Face:
C
C       The first cell of the PGON is defined as a ruled solid between two
C       faces.  The second face is defined as a copy of the first face that is
C       rotated PHIC degrees around the Z-axis.
C
C*        WRITE SET @113,F2..#101,!F1,N1
C-----------------------------------------------
        DO 900 I=1, 500
          BLKSTR(I:I)=' '
  900   CONTINUE
        N1=N1+1
        WRITE(BLKSTR,10030)N1,N1-2,N1-1
        DO 1000 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
 1000   CONTINUE
C*        WRITE SET @100,N2..#145,!F1,!F2
C-----------------------------------------------
        DO 1100 I=1, 500
          BLKSTR(I:I)=' '
 1100   CONTINUE
        N1=N1+1
        WRITE(BLKSTR,10040)N1,N1-3,N1-1
        DO 1200 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
 1200   CONTINUE
C       The rest of the cells are defined as rotated copies of the first cell:
           N2=N1
      DO 110 K=2, NPDV
C*        WRITE SET @100,N3..#101,!N1-1,!NG
C
C-----------------------------------------------
        DO 1300 I=1, 500
          BLKSTR(I:I)=' '
 1300   CONTINUE
        N1=N1+1
        WRITE(BLKSTR,10050)N1,N1-1,NG
        DO 1400 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
 1400   CONTINUE
```

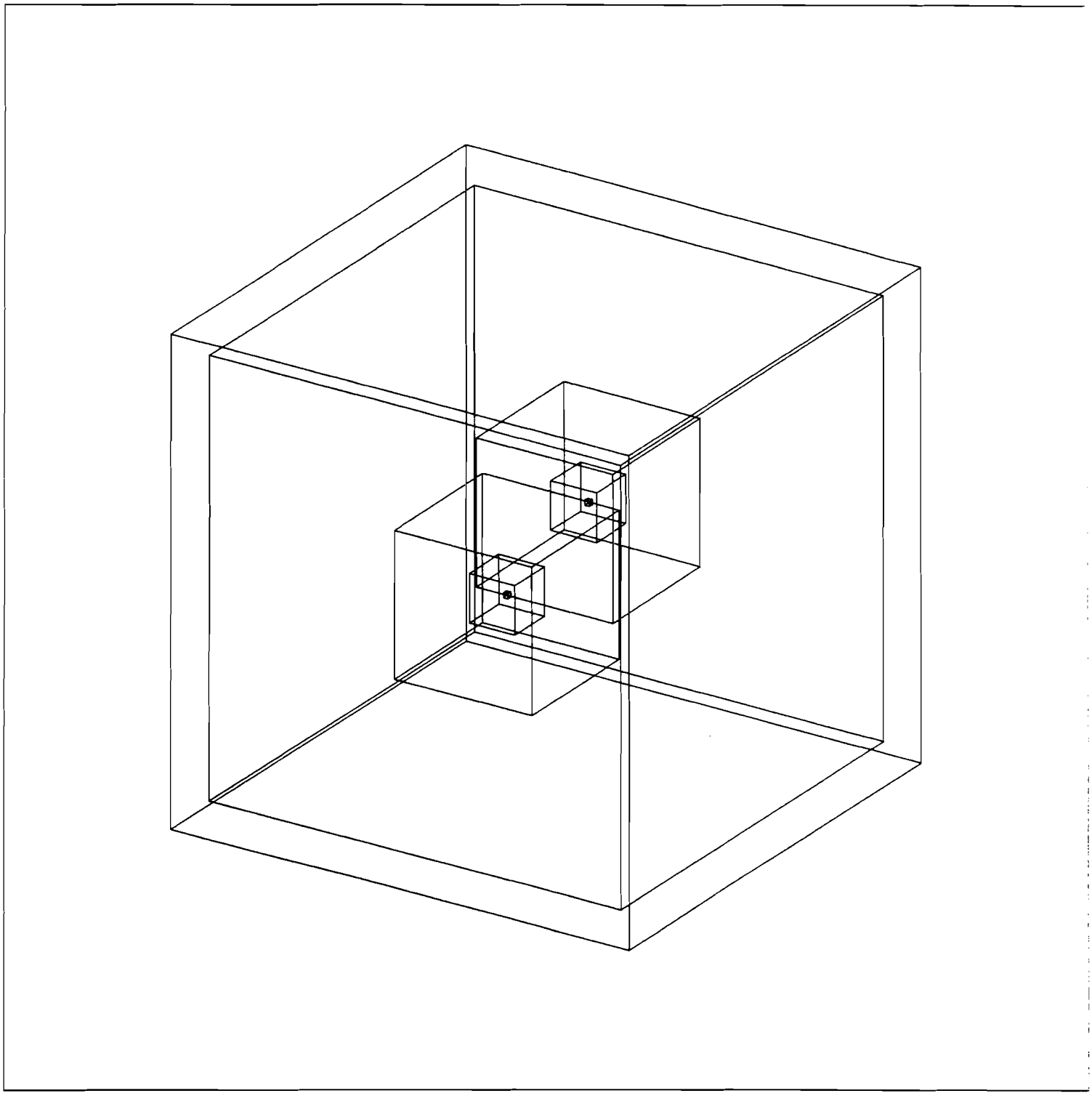
```
 110    CONTINUE
C      The final shape is a Boolean union of the cells
C*       WRITE SET @100,N4..#100,2,!N1-NPDV....!N1-1
C-----------------------------------------------
       DO 1500 I=1, 500
          BLKSTR(I:I)=' '
 1500   CONTINUE
        N1=N1+1
        WRITE(BLKSTR,10070)N1,MATNAM
        DO 1600 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
 1600   CONTINUE
        DO 200 K=N2, N1-1
C-----------------------------------------------
       DO 1700 I=1, 500
          BLKSTR(I:I)=' '
 1700   CONTINUE
        WRITE(BLKSTR,10060)K
        DO 1800 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
 1800   CONTINUE

 200    CONTINUE
C
10005   FORMAT('@103,',I,',:5,2#3,3,2')
10010   FORMAT(',',F,',',F,',',F)
10020   FORMAT('@302,',I,',:5,2#301,',F,',',F,',',F
     +    ,',',F,',',F,',',F,',',F,',',F,',',F)
10030   FORMAT('@113,',I,',:5,2#101,!',I,',!',I)
10040   FORMAT('@100,',I,',:5,2#145,!',I,',!',I)
10050   FORMAT('@100,',I,',:5,2#101,!',I,',!',I)
10060   FORMAT(',!',I)
10070   FORMAT('@100,',I,',:5,2,:9,''',A4,''''#100,2')
C
      NPLACE=J
C
      END
```

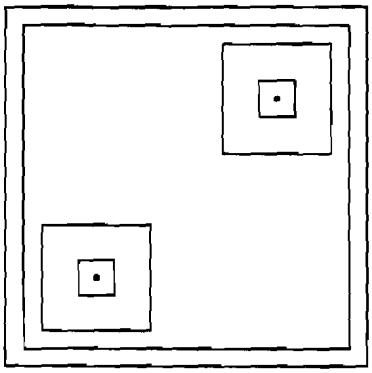
```
*CMZ :          06/11/91  14.43.08  by  Jouko Vuoskoski
*-- Author :  Nils Joar Hoimyr, Jouko Vuoskoski
C
      SUBROUTINE POSIT(PARMAT,XPOSIT,YPOSIT,ZPOSIT,MATNAM,NM)
C     Created  12-JUN-1991   Nils Joar Hoimyr, Jouko Vuoskoski
C
C-----------------------------------------------------------------------
C
*KEEP,SETFIL.
      COMMON/SETFIL/NPLACE,N_UNIT,N1
C
*KEEP,SETCHR.
      COMMON/GEOMTC/BLKSTR  ! this must put in every routine
      CHARACTER*500 BLKSTR   ! no use to be common variable
C
*KEND.
C
C
         DIMENSION  PARMAT(10)
C
C
C-----------------------------------------------------------------------
C
C
C
C *WRITE SET @302,N1,#301,1,0,0,0,1,0,0,0,1,XPOSIT,YPOSIT,XPOSIT
C *WRITE SET @100,N2,:9,'Name'#101,!N1(of the shape),!N1
C
C
C-----------------------------------------------------------------------
C
C      this is for writing without spaces and line length of 80
C
        DO 500 I=1, 500
          BLKSTR(I:I)=' '
  500   CONTINUE
        J=NPLACE
        WRITE(BLKSTR,10010)N1,PARMAT(1),PARMAT(2),PARMAT(3),PARMAT(4),
     +PARMAT(5),PARMAT(6),PARMAT(7),PARMAT(8),PARMAT(9),
     +XPOSIT,YPOSIT,ZPOSIT
        DO 600 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
  600   CONTINUE
        N1=N1+1
        DO 700 I=1, 500
          BLKSTR(I:I)=' '
  700   CONTINUE
        WRITE(BLKSTR,10020)N1,MATNAM,NM,N1-1
        DO 800 I=1, 500
          IF (BLKSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BLKSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
  800   CONTINUE
C
```

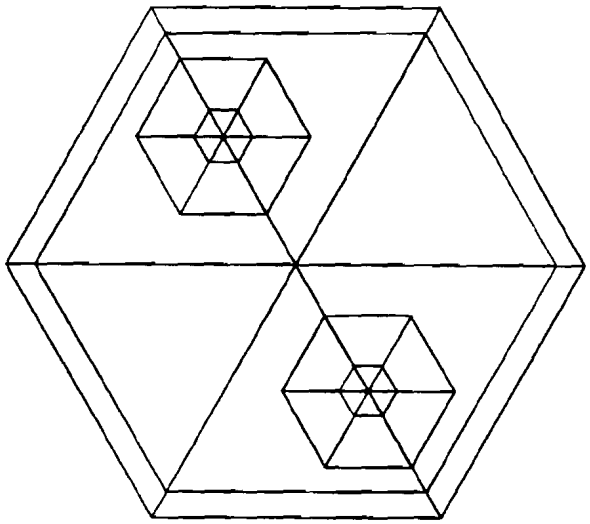
```
10010   FORMAT('@302,',I,'#301,',F,',',F,',',F,',',F,',',F,',',F,
     +',',F,',',F,',',F,',',F,',',F,',',F)
10020   FORMAT('@100,',I,',:5,2,:9,''',A4,'''#101,!',I,',!',I)
C
        NPLACE=J
C
        END
```

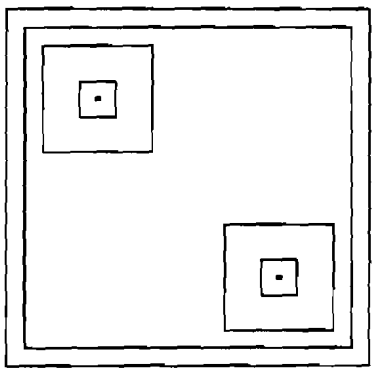
```
*CMZ :          07/11/91  17.51.26  by  Jouko Vuoskoski
*-- Author :   Nils Joar Hoimyr, Jouko Vuoskoski
C
      SUBROUTINE TREE(NCHILD,NBL,NM,MATNAM,IVO2)
C     Created  12-JUN-1991   Nils Joar Hoimyr, Jouko Vuoskoski
C
C     This routine is building the GEANT tree, but it is
C     not taking into account the materials of volumes.
C
C-----------------------------------------------------------------------
C
*KEEP,SETFIL.
      COMMON/SETFIL/NPLACE,N_UNIT,N1
C
*KEEP,SETCHR.
      COMMON/GEOMTC/BLKSTR
      CHARACTER*500 BLKSTR
C
*KEND.
C
C
       INTEGER            NCHILD(1000)
       CHARACTER*10000    BSTR
       CHARACTER*9900     STR
C
C
C-----------------------------------------------------------------------
C
C *WRITE SET @100,N1,:5,2#100,3,!MOTHER,!CHILD1,!CHILD2,...
C *WRITE SET @510,N2,:5,2#510,!N1,!CHILD1,!CHILD2,...
C
C-----------------------------------------------------------------
C
C        write sequence numbers of children into STR
C
        DO 50 I=1, 9900
          STR(I:I)=' '
  50   CONTINUE
        K=1
        J=14
        DO 100 I=1, NBL
          WRITE(STR(K:J),200)NCHILD(I)
          K=K+14
          J=J+14
  100   CONTINUE
  200   FORMAT(',!',I12)
C
C      this is for writing without spaces and line length of 80
C
        DO 500 I=1, 10000
          BSTR(I:I)=' '
  500   CONTINUE
        J=NPLACE
           WRITE(BSTR,10010)N1,NM,STR
        DO 600 I=1, 10000
          IF (BSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
  600   CONTINUE
        N1=N1+1
        DO 700 I=1, 10000
```

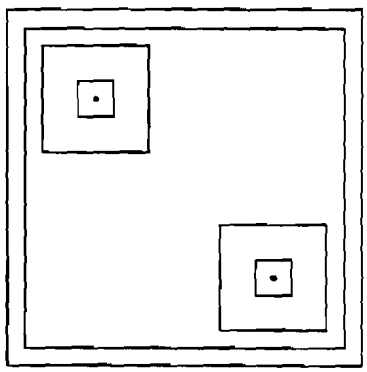
```
          BSTR(I:I)=' '
  700   CONTINUE
         IF (IVO2.EQ.1) THEN
           WRITE(BSTR,10020)N1,MATNAM,N1-1,STR
         ELSE
           WRITE(BSTR,10015)N1,N1-1,STR
         ENDIF
        DO 800 I=1, 10000
          IF (BSTR(I:I) .NE. ' ') THEN
            WRITE(N_UNIT,'(A,$)')BSTR(I:I)
              IF (J .EQ. 80) THEN
                WRITE(N_UNIT,'(/,$)')
                J=0
              END IF
            J=J+1
          END IF
  800   CONTINUE

10010   FORMAT('@100,',I,',:5,2#100,3,!',I,A9900)
10015   FORMAT('@510,',I,',:5,2#510,1,0,0,0,!',I,A9900)
10020   FORMAT('@510,',I,',:9,'''A4'''#510,1,0,0,0,!',I,A9900)
C
        NPLACE=J
C
        END
```

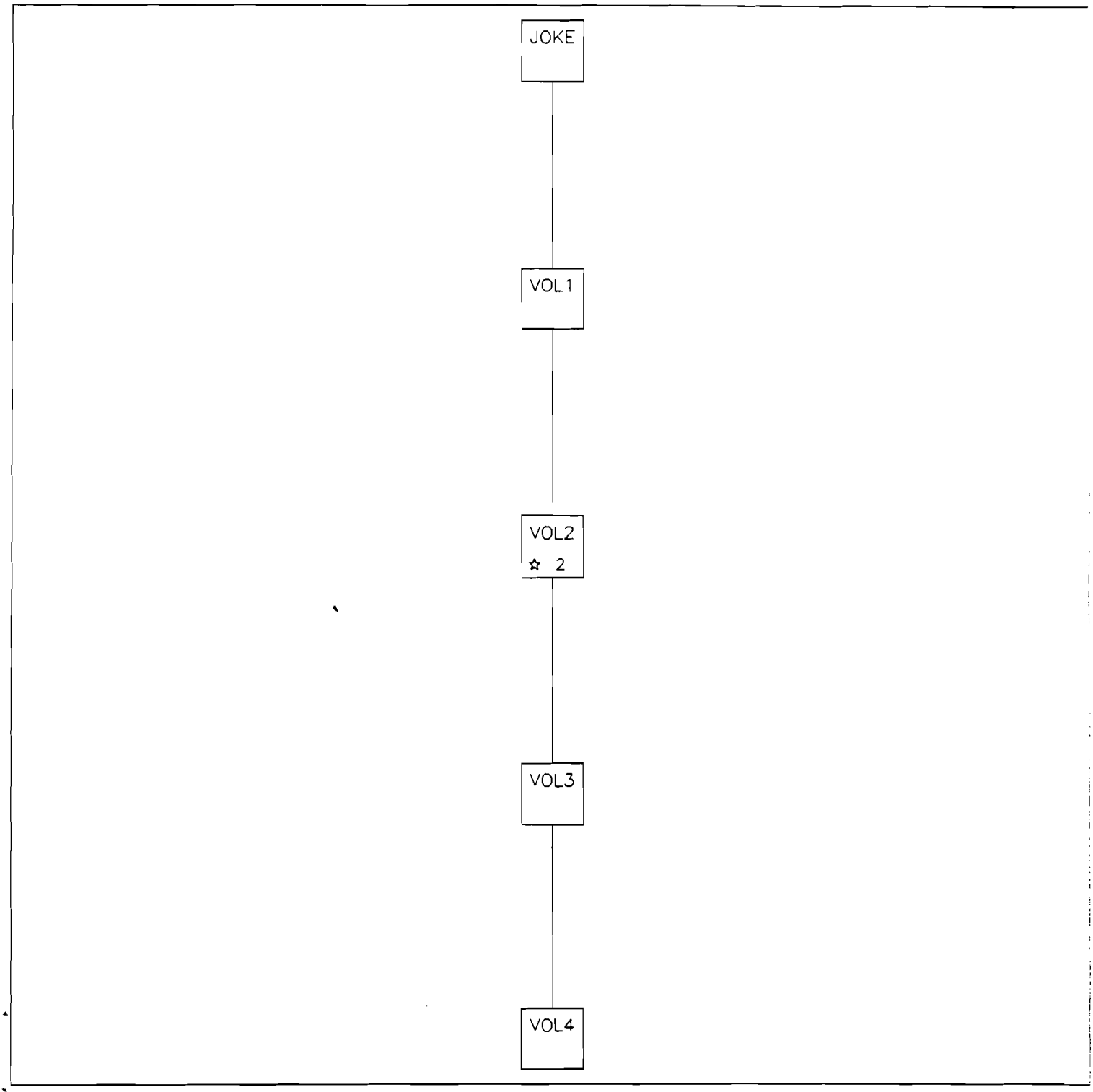


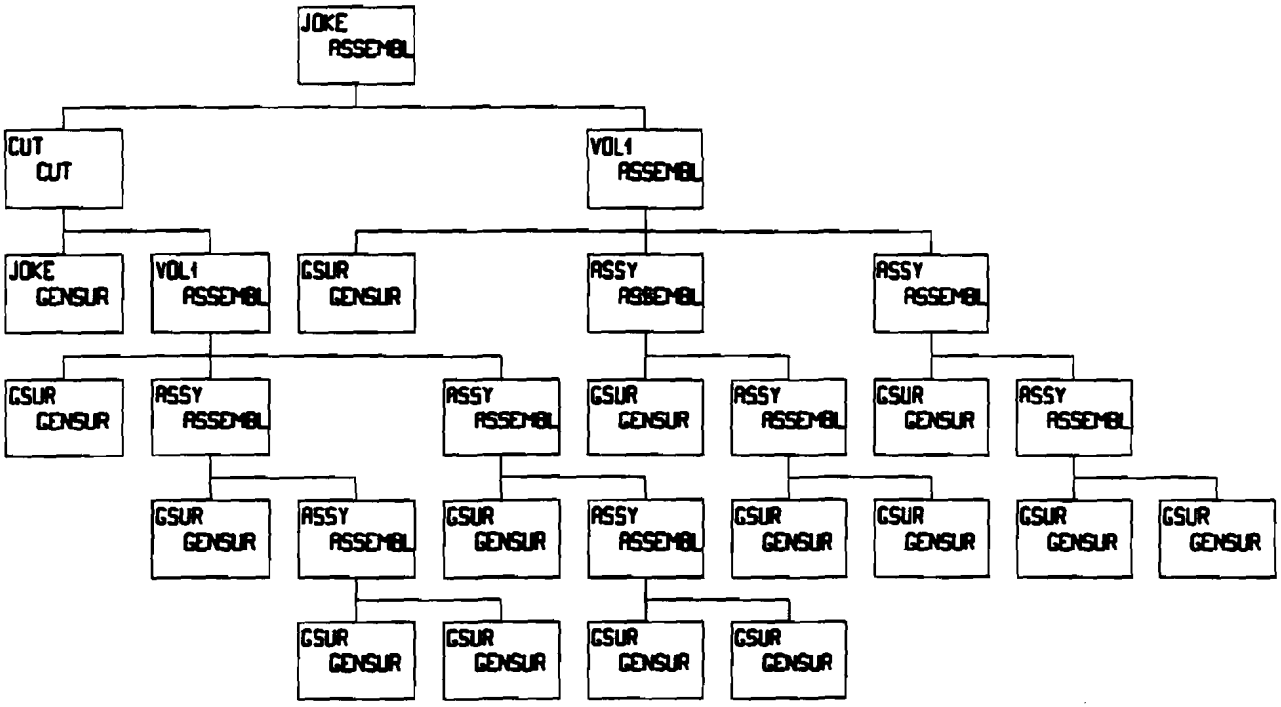






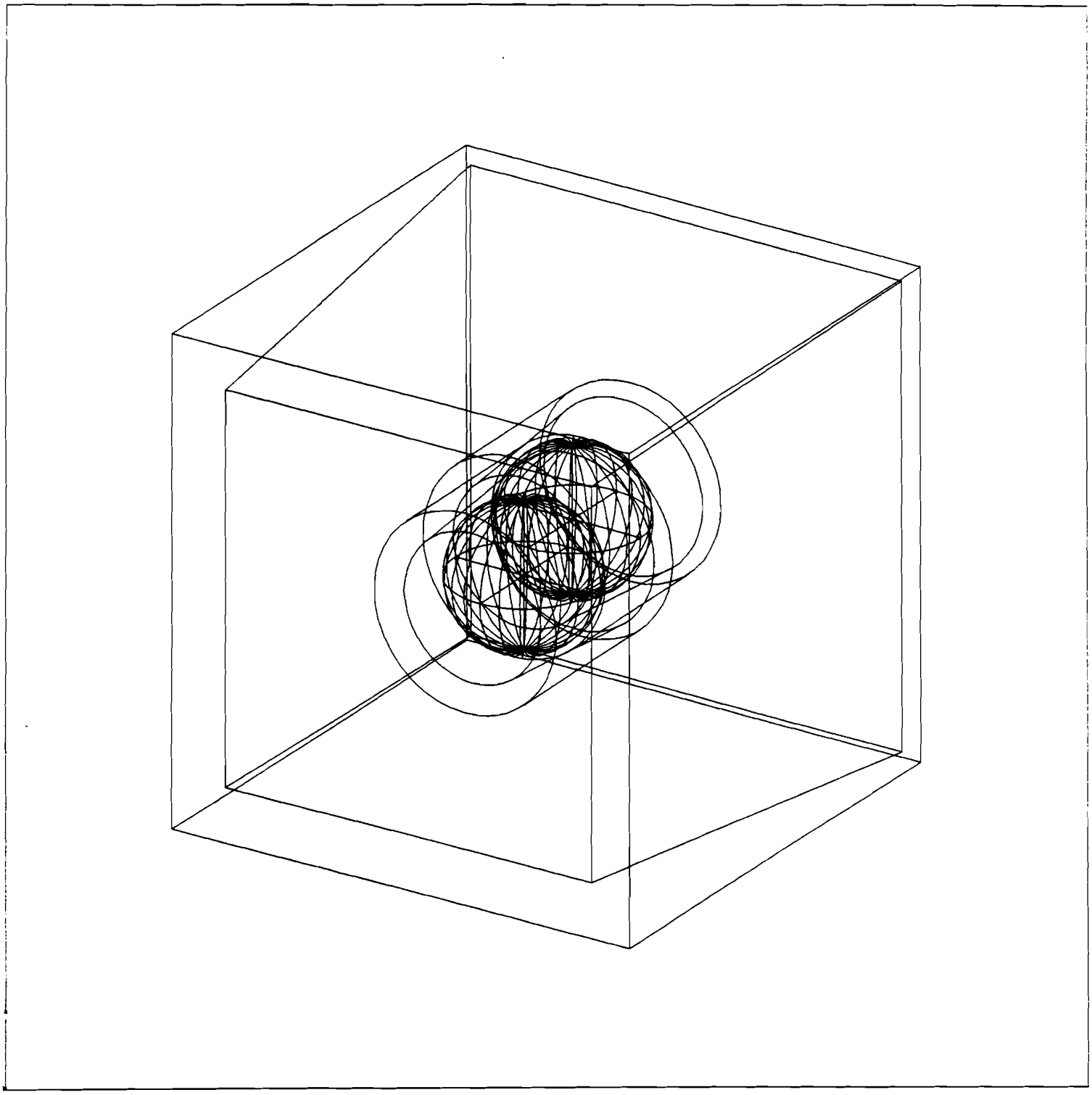


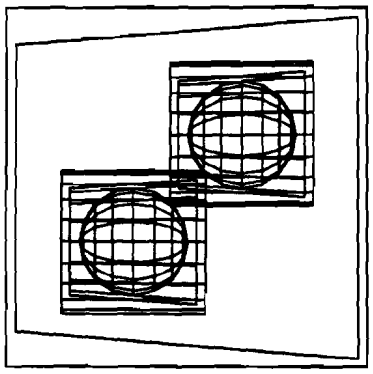

JOKE
VOL1
VOL2
☆ 2
VOL3
VOL4

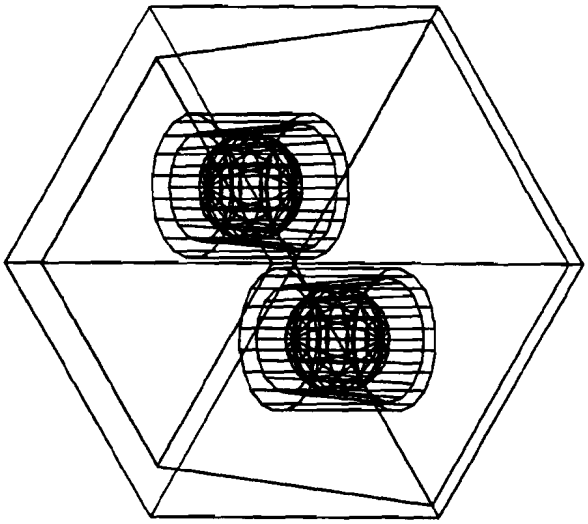

JOKE ASSEMBL
CUT CUT
VOL1 ASSEMBL
JOKE GENSUR
VOL1 ASSEMBL
GSUR GENSUR
ASSY ASSEMBL
ASSY ASSEMBL
GSUR GENSUR
ASSY ASSEMBL
ASSY ASSEMBL
GSUR GENSUR
ASSY ASSEMBL
GSUR GENSUR
ASSY ASSEMBL
GSUR GENSUR
ASSY ASSEMBL
GSUR GENSUR
GSUR GENSUR
GSUR GENSUR
GSUR GENSUR
GSUR GENSUR
GSUR GENSUR
GSUR GENSUR
GSUR GENSUR
GSUR GENSUR

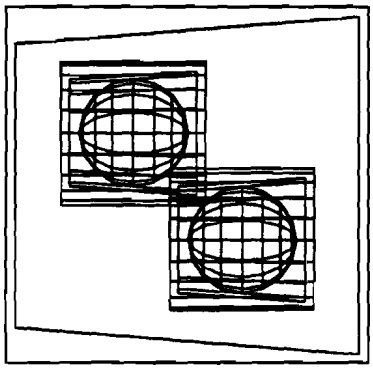
```
*CMZ :          06/11/91  17.45.23  by  Jouko Vuoskoski
*-- Author :    Federico Carminati   24/05/91
      SUBROUTINE UGEOM
*
*************************************************************************
*                                                                       *
*             Routine to define the geometry of the set-up.             *
*                                                                       *
*************************************************************************
*
*             Definition of the mother (box)
*
      REAL PAR(10)
*
*
      PAR(1)=100.
      PAR(2)=100.
      PAR(3)=100.
      CALL GSVOLU('JOKE','BOX ',1,PAR,3,IVOLU)
*
*             Definition of the daughters
*
      PAR(1)=90.
      PAR(2)=90.
      PAR(3)=90.
      CALL GSVOLU('VOL1','BOX ',1,PAR,3,IVOLU)
      CALL GSPOS('VOL1',1,'JOKE',0.0,0.0,0.0,0,'ONLY')
*
*
      PAR(1)=30.
      PAR(2)=30.
      PAR(3)=30.
      CALL GSVOLU('VOL2','BOX ',1,PAR,3,IVOLU)
      CALL GSPOS('VOL2',1,'VOL1',50.0,50.0,50.0,0,'ONLY')
      CALL GSPOS('VOL2',2,'VOL1',-50.0,-50.0,-50.0,0,'ONLY')
*
      PAR(1)=10.
      PAR(2)=10.
      PAR(3)=10.
      CALL GSVOLU('VOL3','BOX ',1,PAR,3,IVOLU)
      CALL GSPOS('VOL3',1,'VOL2',0.0,0.0,0.0,0,'ONLY')
*
      PAR(1)=1.
      PAR(2)=1.
      PAR(3)=1.
      CALL GSVOLU('VOL4','BOX ',1,PAR,3,IVOLU)
      CALL GSPOS('VOL4',1,'VOL3',0.0,0.0,0.0,0,'ONLY')
*
*             Close geometry banks. Mandatory system routine.
*
      CALL GGCLOS
*
      END
```

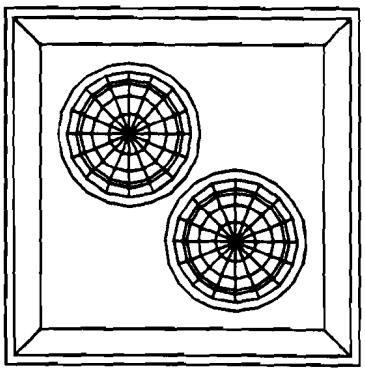
@9900#9900,'Z68-300.89.06','2.0','GEANT-SET','1.0','CERN','','',''
#9901,1991,11,11,16,49,47
@9901,1,'JOKE   '#9905,3,1.
,1,0.001,0,-300.,300.,-300.,300.,-300.,300.
#9920,:12,0.005,:14,0,:15,0
@301,2#301,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.
@50,3,:5,2#60,2.0000000000000000,2.0000000000000000,2.0000000000000000@302,4#317
,-1.0000000,-1.0000000,-1.0000000@100,5,:5,2,:9,'VOL4'#101,!3,!4@302,6#301,1.000
0000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.000
0000,0.0000000,0.0000000,0.0000000@100,7,:5,2,:9,'VOL4'#101,!5,!6@50,8,:5,2#60,2
0.0000000000000000,20.0000000000000000,20.0000000000000000@302,9#317,-10.0000000
,-10.0000000,-10.0000000@100,10,:5,2,:9,'VOL3'#101,!8,!9@100,11,:5,2#100,3,!10,!
7@510,12,:5,2#510,1,0,0,0,!10,!7@302,13#301,1.0000000,0.0000000,0.0000000,0.0000
000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000
000@100,14,:5,2,:9,'VOL3'#101,!12,!13@50,15,:5,2#60,60.0000000000000000,60.00000
00000000000,60.0000000000000000@302,16#317,-30.0000000,-30.0000000,-30.0000000@1
00,17,:5,2,:9,'VOL2'#101,!15,!16@100,18,:5,2#100,3,!17,!14@510,19,:5,2#510,1,0,0
,0,!17,!14@302,20#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.000000
0,0.0000000,0.0000000,1.0000000,50.0000000,50.0000000,50.0000000@100,21,:5,2,:9,
'VOL2'#101,!19,!20@302,22#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,
0.0000000,0.0000000,0.0000000,1.0000000,-50.0000000,-50.0000000,-50.0000000@100,
23,:5,2,:9,'VOL2'#101,!19,!22@50,24,:5,2#60,180.0000000000000000,180.00000000000
00000,180.0000000000000000@302,25#317,-90.0000000,-90.0000000,-90.0000000@100,26
,:5,2,:9,'VOL1'#101,!24,!25@100,27,:5,2#100,3,!26,!21,!23@510,28,:5,2#510,1,0,0,
0,!26,!21,!23@302,29#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.000
0000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000@100,30,:5,2,:9,
'VOL1'#101,!28,!29@50,31,:5,2#60,200.0000000000000000,200.0000000000000000,200.0
0000000000000000@302,32#317,-100.0000000,-100.0000000,-100.0000000@100,33,:5,2,:9
,'JOKE'#101,!31,!32@100,34,:5,2#100,3,!33,!30@510,35,:9,'JOKE'#510,1,0,0,0,!34,!
30@9998,36@9999,38

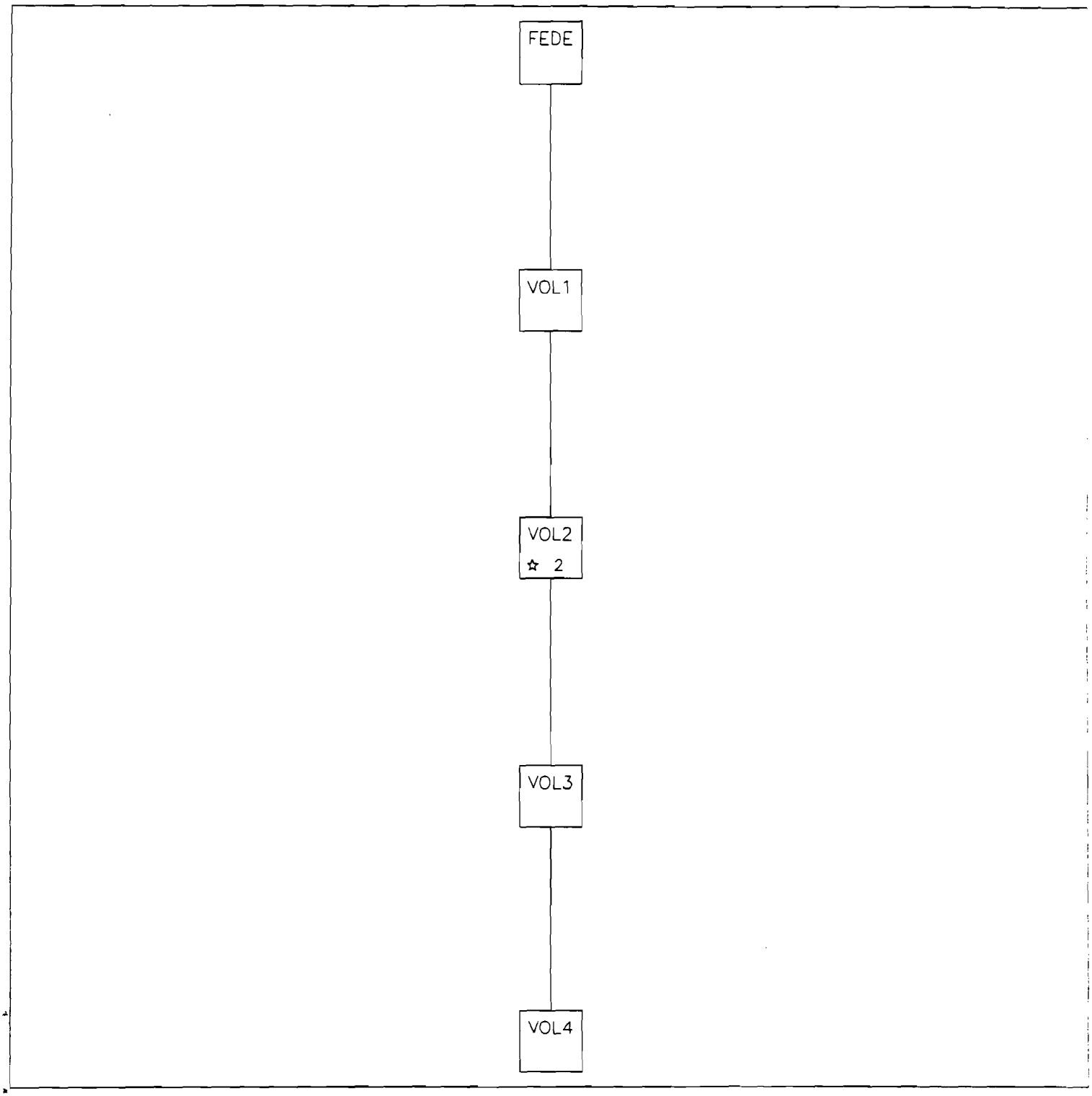


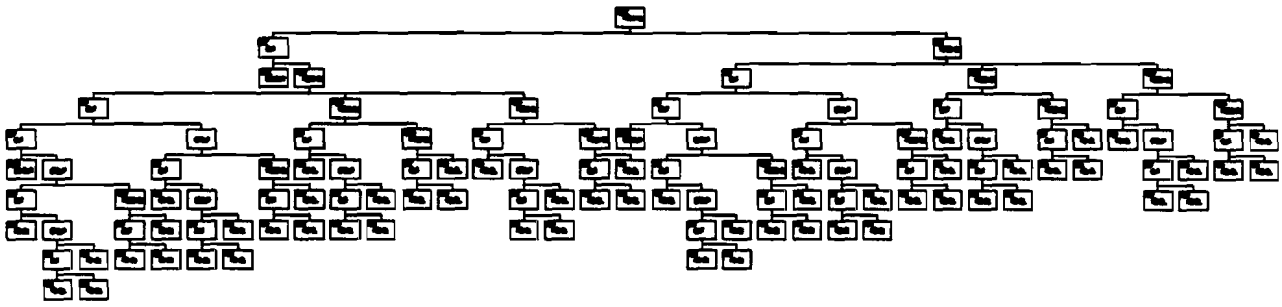






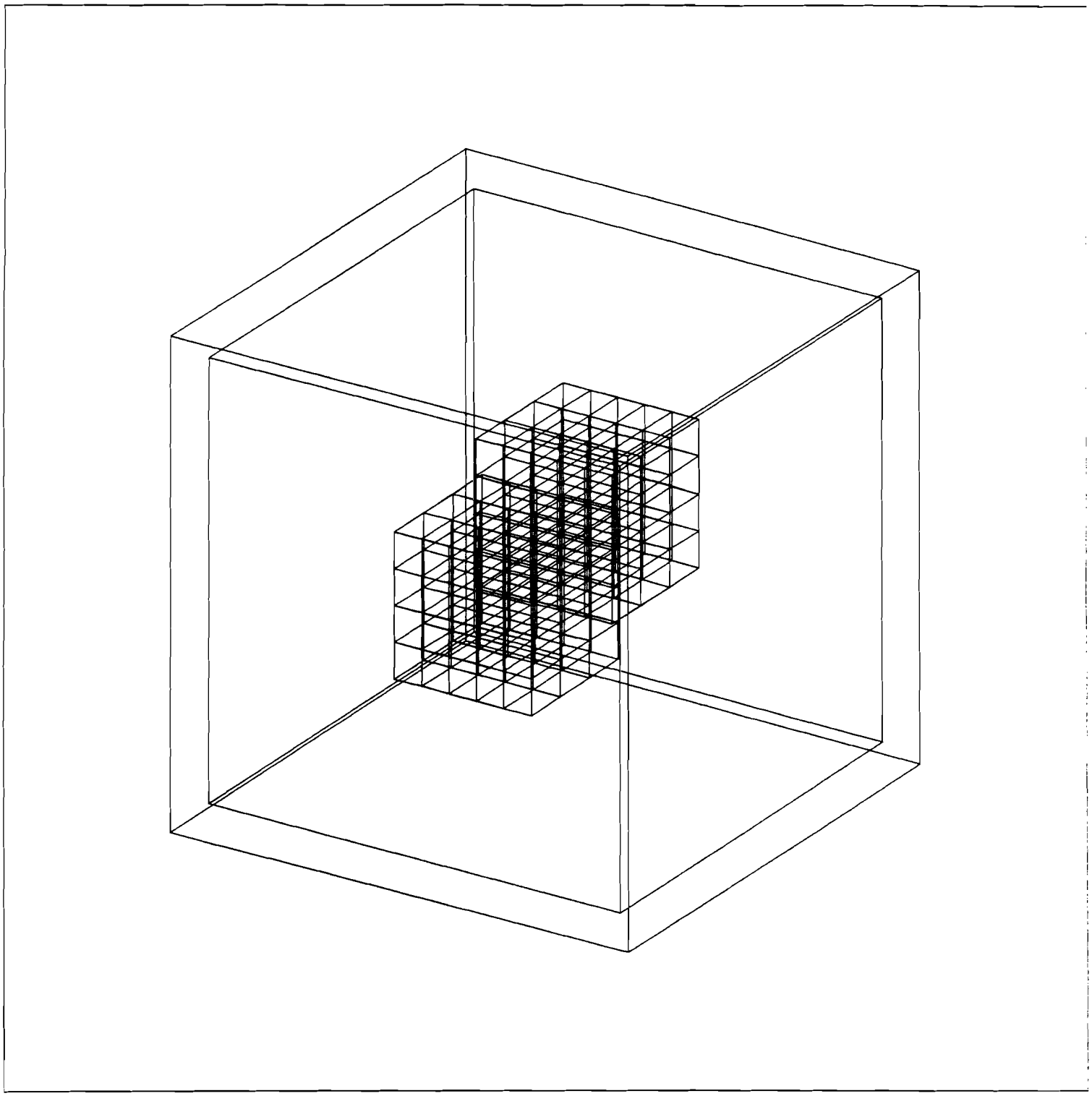


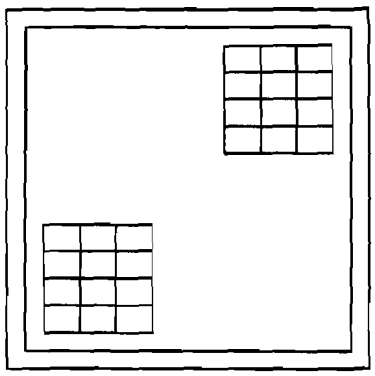

FEDE
VOL1
VOL2
☆ 2
VOL3
VOL4

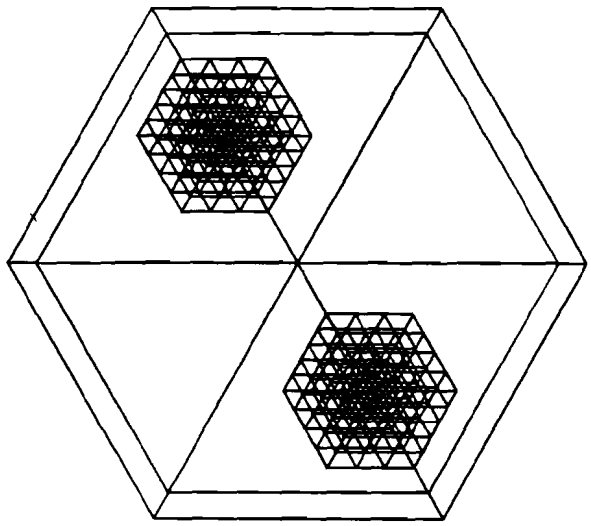


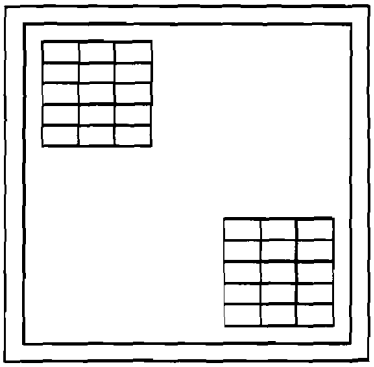
```
*CMZ :          06/11/91  17.51.34  by  Jouko Vuoskoski
*-- Author :    Federico Carminati   24/05/91
      SUBROUTINE UGEOM
*
*************************************************************************
*                                                                       *
*             Routine to define the geometry of the set-up.             *
*                                                                       *
*************************************************************************
*
*             Definition of the mother (box)
*
      REAL PAR(10)
*
*
      PAR(1)=100.
      PAR(2)=100.
      PAR(3)=100.
      CALL GSVOLU('FEDE','BOX ',1,PAR,3,IVOLU)
*
*             Definition of the daughters
*
      PAR(1)=95.
      PAR(2)=80.
      PAR(3)=95.
      PAR(4)=80.
      PAR(5)=95.
      CALL GSVOLU('VOL1','TRD2',1,PAR,5,IVOLU)
      CALL GSPOS('VOL1',1,'FEDE',0.0,0.0,0.0,0,'ONLY')
*
*
      PAR(1)=0.
      PAR(2)=40.
      PAR(3)=40.
      CALL GSVOLU('VOL2','TUBE',1,PAR,3,IVOLU)
      CALL GSPOS('VOL2',1,'VOL1',30.0,30.0,30.0,0,'ONLY')
      CALL GSPOS('VOL2',2,'VOL1',-30.0,-30.0,-30.0,0,'ONLY')
*
      PAR(1)=35.
      PAR(2)=0.
      PAR(3)=35.
      PAR(4)=0.
      PAR(5)=30.
      CALL GSVOLU('VOL3','CONE',1,PAR,5,IVOLU)
      CALL GSPOS('VOL3',1,'VOL2',0.0,0.0,0.0,0,'ONLY')
*
      PAR(1)=0.
      PAR(2)=30.
      PAR(3)=0.
      PAR(4)=0.
      PAR(5)=0.
      PAR(6)=0.
      CALL GSVOLU('VOL4','SPHE',1,PAR,6,IVOLU)
      CALL GSPOS('VOL4',1,'VOL3',0.0,0.0,0.0,0,'ONLY')
*
*             Close geometry banks. Mandatory system routine.
*
      CALL GGCLOS
*
      END
```

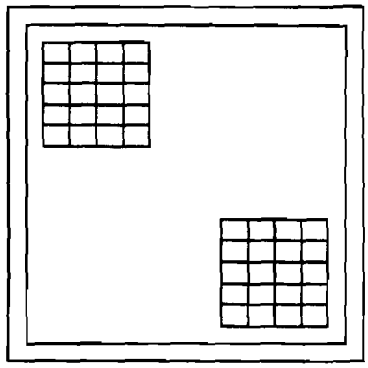
@9900#9900,'Z68-300.89.06','2.0','GEANT-SET','1.0','CERN','','',''
#9901,1991,11,11,16,42,05
@9901,1,'FEDE   '#9905,3,1.
,1,0.001,0,-300.,300.,-300.,300.,-300.,300.
#9920,:12,0.005,:14,0,:15,0
@301,2#301,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.
@50,3,:5,2,:9,'VOL4'#34,30.0000000,,,,,0.0000000@302,4#301,1.0000000,0.0000000,0
.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0
.0000000,0.0000000@100,5,:5,2,:9,'VOL4'#101,!3,!4@103,6,:5,2#3,3,2,0.0000000,0.0
000000,-35.0000000,0.0000000,0.0000000,35.0000000,30.0000000,0.0000000,35.000000
0,35.0000000,0.0000000,-35.0000000@100,7,:5,2,:9,'VOL3'#144,!6,0.0000000,360.000
0000@100,8,:5,2#100,3,!7,!5@510,9,:5,2#510,1,0,0,0,!8,!5@302,10#301,1.0000000,0.
0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.
0000000,0.0000000,0.0000000@100,11,:5,2,:9,'VOL3'#101,!9,!10@50,12,:5,2#32,40.00
00000,80.0000000,0.0000000,360.0000000,0.0000000@302,13#317,-0.0000000,0.0000000
,-40.0000000@100,14,:5,2,:9,'VOL2'#101,!12,!13@100,15,:5,2#100,3,!14,!11@510,16,
:5,2#510,1,0,0,0,!15,!11@302,17#301,1.0000000,0.0000000,0.0000000,0.0000000,1.00
00000,0.0000000,0.0000000,0.0000000,1.0000000,30.0000000,30.0000000,30.0000000@1
00,18,:5,2,:9,'VOL2'#101,!16,!17@302,19#301,1.0000000,0.0000000,0.0000000,0.0000
000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,-30.0000000,-30.0000000,-3
0.0000000@100,20,:5,2,:9,'VOL2'#101,!16,!19@103,21,:5,2#3,3,2,-95.0000000,-95.00
00000,-95.0000000,95.0000000,-95.0000000,-95.0000000,95.0000000,95.0000000,-95.0
000000,-95.0000000,95.0000000,-95.0000000@103,22,:5,2#3,3,2,-80.0000000,-80.0000
000,95.0000000,80.0000000,-80.0000000,95.0000000,80.0000000,80.0000000,95.000000
0,-80.0000000,80.0000000,95.0000000@100,23,:5,2,:9,'VOL1'#145,!21,!22@100,24,:5,
2#100,3,!23,!18,!20@510,25,:5,2#510,1,0,0,0,!24,!18,!20@302,26#301,1.0000000,0.0
000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0
000000,0.0000000,0.0000000@100,27,:5,2,:9,'VOL1'#101,!25,!26@50,28,:5,2#60,200.0
000000000000000,200.0000000000000000,200.0000000000000000@302,29#317,-100.000000
0,-100.0000000,-100.0000000@100,30,:5,2,:9,'FEDE'#101,!28,!29@100,31,:5,2#100,3,
!30,!27@510,32,:9,'FEDE'#510,1,0,0,0,!31,!27@9998,33@9999,35

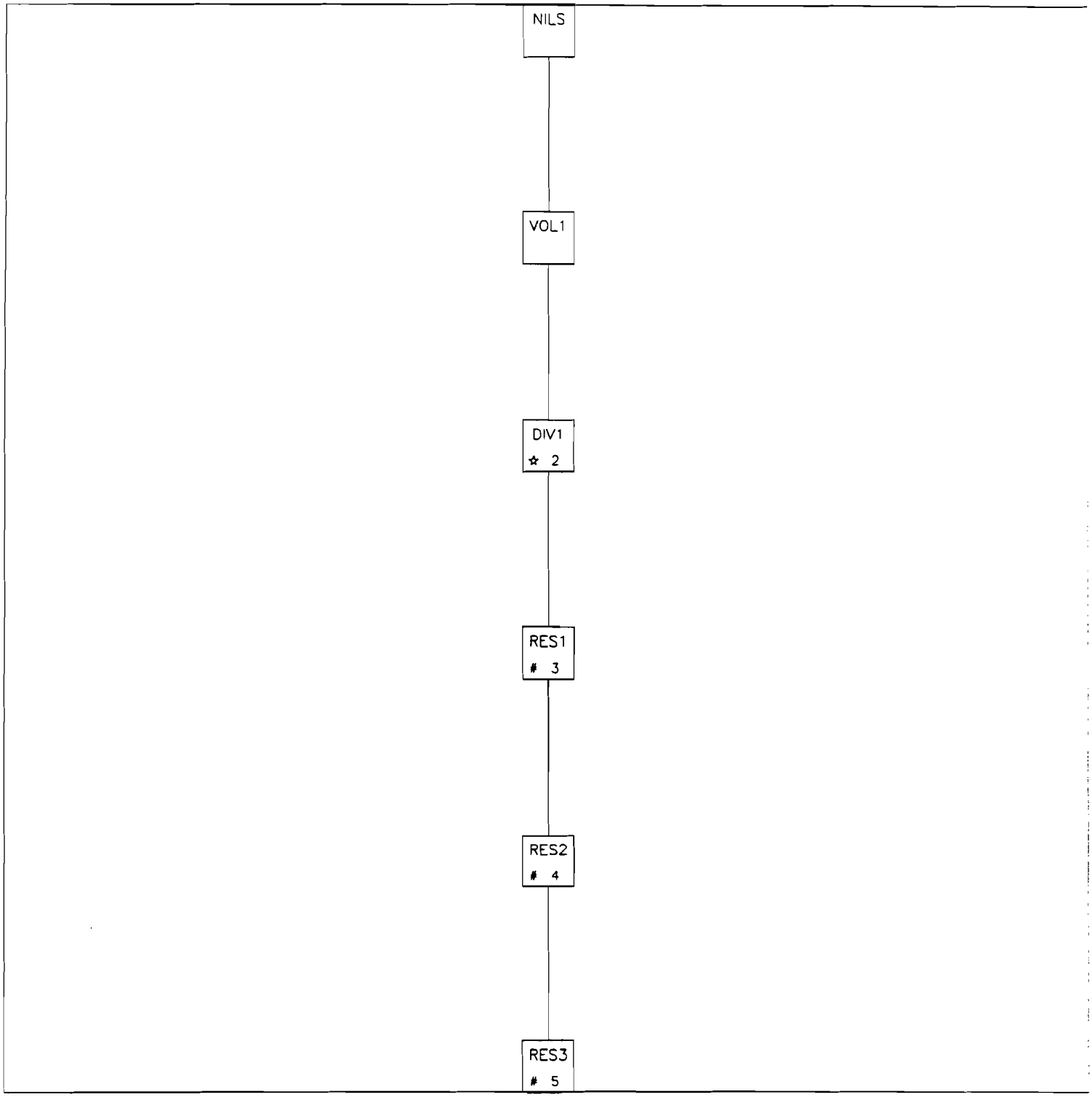








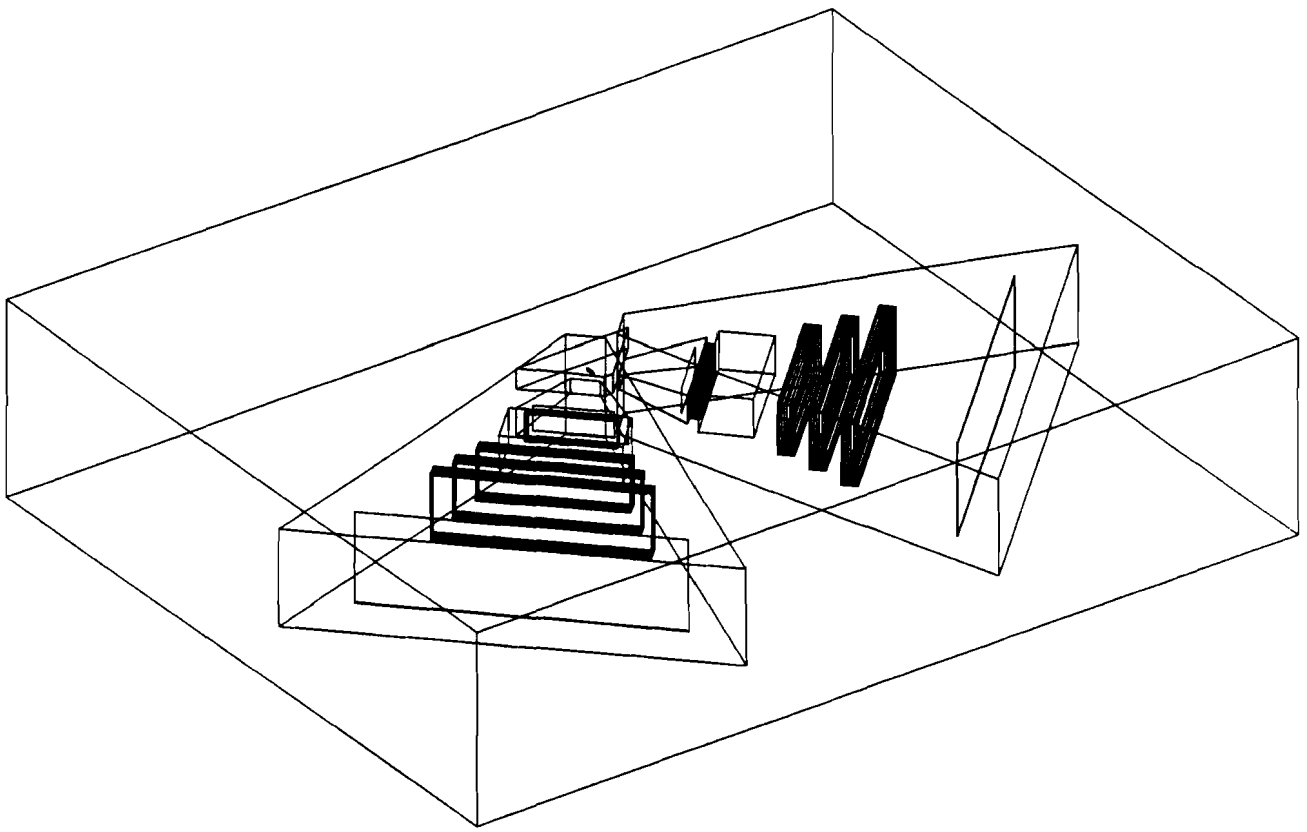


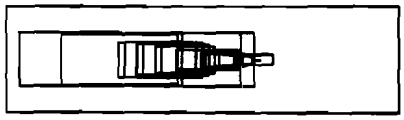

NILS
VOL1
DIV1
☆ 2
RES1
3
RES2
4
RES3
5

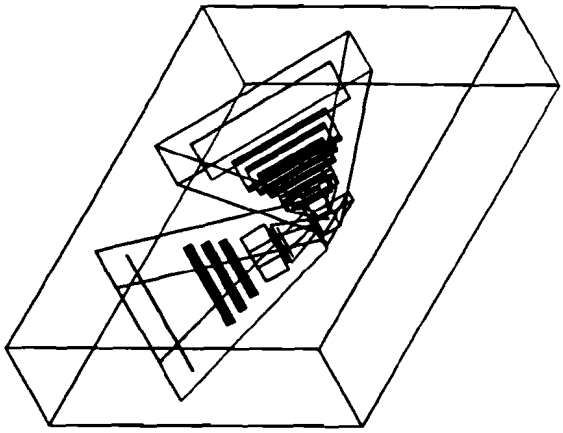
```
*CMZ :          06/11/91  17.48.13  by  Jouko Vuoskoski
*-- Author :    Federico Carminati   24/05/91
      SUBROUTINE UGEOM
*
*************************************************************************
*                                                                       *
*             Routine to define the geometry of the set-up.             *
*                                                                       *
*************************************************************************
*
*             Definition of the mother (box)
*
      REAL PAR(10)
*
*
      PAR(1)=100.
      PAR(2)=100.
      PAR(3)=100.
      CALL GSVOLU('NILS','BOX ',1,PAR,3,IVOLU)
*
*             Definition of the daughters
*
      PAR(1)=90.
      PAR(2)=90.
      PAR(3)=90.
      CALL GSVOLU('VOL1','BOX ',1,PAR,3,IVOLU)
      CALL GSPOS('VOL1',1,'NILS',0.0,0.0,0.0,0,'ONLY')
*
*
      PAR(1)=30.
      PAR(2)=30.
      PAR(3)=30.
      CALL GSVOLU('DIV1','BOX ',1,PAR,3,IVOLU)
      CALL GSPOS('DIV1',1,'VOL1',50.0,50.0,50.0,0,'ONLY')
      CALL GSPOS('DIV1',2,'VOL1',-50.0,-50.0,-50.0,0,'ONLY')
*
      CALL GSDVN('RES1','DIV1',3,3)
      CALL GSDVN('RES2','RES1',4,2)
      CALL GSDVN('RES3','RES2',5,1)
*
*
*             Close geometry banks. Mandatory system routine.
*
      CALL GGCLOS
*
      END
```

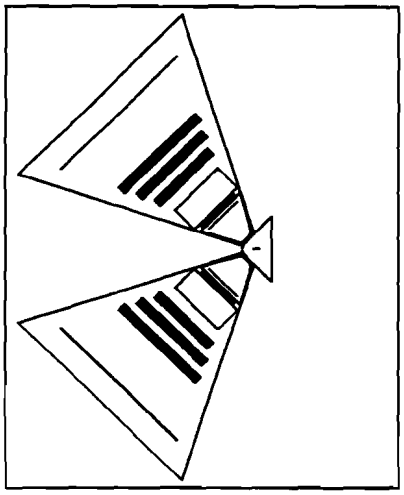
```
@9900#9900,'Z68-300.89.06','2.0','GEANT-SET','1.0','CERN','','',''
#9901,1991,11,12,12,02,34
@9901,1,'NILS      '#9905,3,1.
,1,0.001,0,-300.,300.,-300.,300.,-300.,300.
#9920,:12,0.005,:14,0,:15,0
@301,2#301,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.
@50,3,:5,2#60,12.0000000000000000,15.0000000000000000,20.0000000000000000@302,4#
317,-6.0000000,-7.5000000,-10.0000000@100,5,:5,2,:9,'RES3'#101,!3,!4@50,6,:5,2#6
0,12.0000000000000000,15.0000000000000000,20.0000000000000000@302,7#317,-6.00000
00,-7.5000000,-10.0000000@100,8,:5,2,:9,'RES3'#101,!6,!7@50,9,:5,2#60,12.0000000
000000000,15.0000000000000000,20.0000000000000000@302,10#317,-6.0000000,-7.50000
00,-10.0000000@100,11,:5,2,:9,'RES3'#101,!9,!10@50,12,:5,2#60,12.000000000000000
0,15.0000000000000000,20.0000000000000000@302,13#317,-6.0000000,-7.5000000,-10.0
000000@100,14,:5,2,:9,'RES3'#101,!12,!13@50,15,:5,2#60,12.0000000000000000,15.00
00000000000000,20.0000000000000000@302,16#317,-6.0000000,-7.5000000,-10.0000000@
100,17,:5,2,:9,'RES3'#101,!15,!16@302,18#301,1.0000000,0.0000000,0.0000000,0.000
0000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,-24.0000000,0.0000000,0.0
000000@100,19,:5,2,:9,'RES3'#101,!5,!18@302,20#301,1.0000000,0.0000000,0.0000000
,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,-12.0000000,0.00000
00,0.0000000@100,21,:5,2,:9,'RES3'#101,!8,!20@302,22#301,1.0000000,0.0000000,0.0
000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0
000000,0.0000000@100,23,:5,2,:9,'RES3'#101,!11,!22@302,24#301,1.0000000,0.000000
0,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,12.00000
00,0.0000000,0.0000000@100,25,:5,2,:9,'RES3'#101,!14,!24@302,26#301,1.0000000,0.
0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,24
.0000000,0.0000000,0.0000000@100,27,:5,2,:9,'RES3'#101,!17,!26@50,28,:5,2#60,60.
0000000000000000,15.0000000000000000,20.0000000000000000@302,29#317,-30.0000000,
-7.5000000,-10.0000000@100,30,:5,2,:9,'RES2'#101,!28,!29@100,31,:5,2#100,3,!30,!
19,!21,!23,!25,!27@510,32,:5,2#510,1,0,0,0,!31,!19,!21,!23,!25,!27@50,33,:5,2#60
,60.0000000000000000,15.0000000000000000,20.0000000000000000@302,34#317,-30.0000
000,-7.5000000,-10.0000000@100,35,:5,2,:9,'RES2'#101,!33,!34@100,36,:5,2#100,3,!
35,!19,!21,!23,!25,!27@510,37,:5,2#510,1,0,0,0,!36,!19,!21,!23,!25,!27@50,38,:5,
2#60,60.0000000000000000,15.0000000000000000,20.0000000000000000@302,39#317,-30.
0000000,-7.5000000,-10.0000000@100,40,:5,2,:9,'RES2'#101,!38,!39@100,41,:5,2#100
,3,!40,!19,!21,!23,!25,!27@510,42,:5,2#510,1,0,0,0,!41,!19,!21,!23,!25,!27@50,43
,:5,2#60,60.0000000000000000,15.0000000000000000,20.0000000000000000@302,44#317,
-30.0000000,-7.5000000,-10.0000000@100,45,:5,2,:9,'RES2'#101,!43,!44@100,46,:5,2
#100,3,!45,!19,!21,!23,!25,!27@510,47,:5,2#510,1,0,0,0,!46,!19,!21,!23,!25,!27@3
02,48#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,
0.0000000,1.0000000,0.0000000,-22.5000000,0.0000000@100,49,:5,2,:9,'RES2'#101,!3
2,!48@302,50#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0
000000,0.0000000,1.0000000,0.0000000,-7.5000000,0.0000000@100,51,:5,2,:9,'RES2'#
101,!37,!50@302,52#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.00000
00,0.0000000,0.0000000,1.0000000,0.0000000,7.5000000,0.0000000@100,53,:5,2,:9,'R
ES2'#101,!42,!52@302,54#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.
0000000,0.0000000,0.0000000,1.0000000,0.0000000,22.5000000,0.0000000@100,55,:5,2
,:9,'RES2'#101,!47,!54@50,56,:5,2#60,60.0000000000000000,60.0000000000000000,20.
0000000000000000@302,57#317,-30.0000000,-30.0000000,-10.0000000@100,58,:5,2,:9,'
RES1'#101,!56,!57@100,59,:5,2#100,3,!58,!49,!51,!53,!55@510,60,:5,2#510,1,0,0,0,
!59,!49,!51,!53,!55@50,61,:5,2#60,60.0000000000000000,60.0000000000000000,20.000
0000000000000@302,62#317,-30.0000000,-30.0000000,-10.0000000@100,63,:5,2,:9,'RES
1'#101,!61,!62@100,64,:5,2#100,3,!63,!49,!51,!53,!55@510,65,:5,2#510,1,0,0,0,!64
,!49,!51,!53,!55@50,66,:5,2#60,60.0000000000000000,60.0000000000000000,20.000000
0000000000@302,67#317,-30.0000000,-30.0000000,-10.0000000@100,68,:5,2,:9,'RES1'#
101,!66,!67@100,69,:5,2#100,3,!68,!49,!51,!53,!55@510,70,:5,2#510,1,0,0,0,!69,!4
9,!51,!53,!55@302,71#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.000
0000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-20.0000000@100,72,:5,2,:
9,'RES1'#101,!60,!71@302,73#301,1.0000000,0.0000000,0.0000000,0.0000000,1.000000
0,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000@100,74,:
5,2,:9,'RES1'#101,!65,!73@302,75#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0
000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,20.0000000@10
0,76,:5,2,:9,'RES1'#101,!70,!75@50,77,:5,2#60,60.0000000000000000,60.00000000000
00000,60.0000000000000000@302,78#317,-30.0000000,-30.0000000,-30.0000000@100,79,
:5,2,:9,'DIV1'#101,!77,!78@100,80,:5,2#100,3,!79,!72,!74,!76@510,81,:5,2#510,1,0
,0,0,!80,!72,!74,!76@302,82#301,1.0000000,0.0000000,0.0000000,0.0000000,1.000000
0,0.0000000,0.0000000,0.0000000,1.0000000,50.0000000,50.0000000,50.0000000@100,8
```

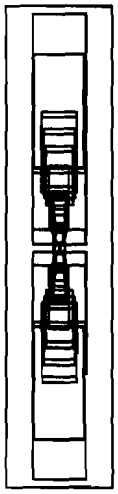
```
3,:5,2,:9,'DIV1'#101,!81,!82@302,84#301,1.0000000,0.0000000,0.0000000,0.0000000,
1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,-50.0000000,-50.0000000,-50.00
00000@100,85,:5,2,:9,'DIV1'#101,!81,!84@50,86,:5,2#60,180.0000000000000000,180.0
000000000000000,180.0000000000000000@302,87#317,-90.0000000,-90.0000000,-90.0000
000@100,88,:5,2,:9,'VOL1'#101,!86,!87@100,89,:5,2#100,3,!88,!83,!85@510,90,:5,2#
510,1,0,0,0,!89,!83,!85@302,91#301,1.0000000,0.0000000,0.0000000,0.0000000,1.000
0000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000@100,9
2,:5,2,:9,'VOL1'#101,!90,!91@50,93,:5,2#60,200.0000000000000000,200.000000000000
0000,200.0000000000000000@302,94#317,-100.0000000,-100.0000000,-100.0000000@100,
95,:5,2,:9,'NILS'#101,!93,!94@100,96,:5,2#100,3,!95,!92@510,97,:9,'NILS'#510,1,0
,0,0,!96,!92@9998,98@9999,100
```

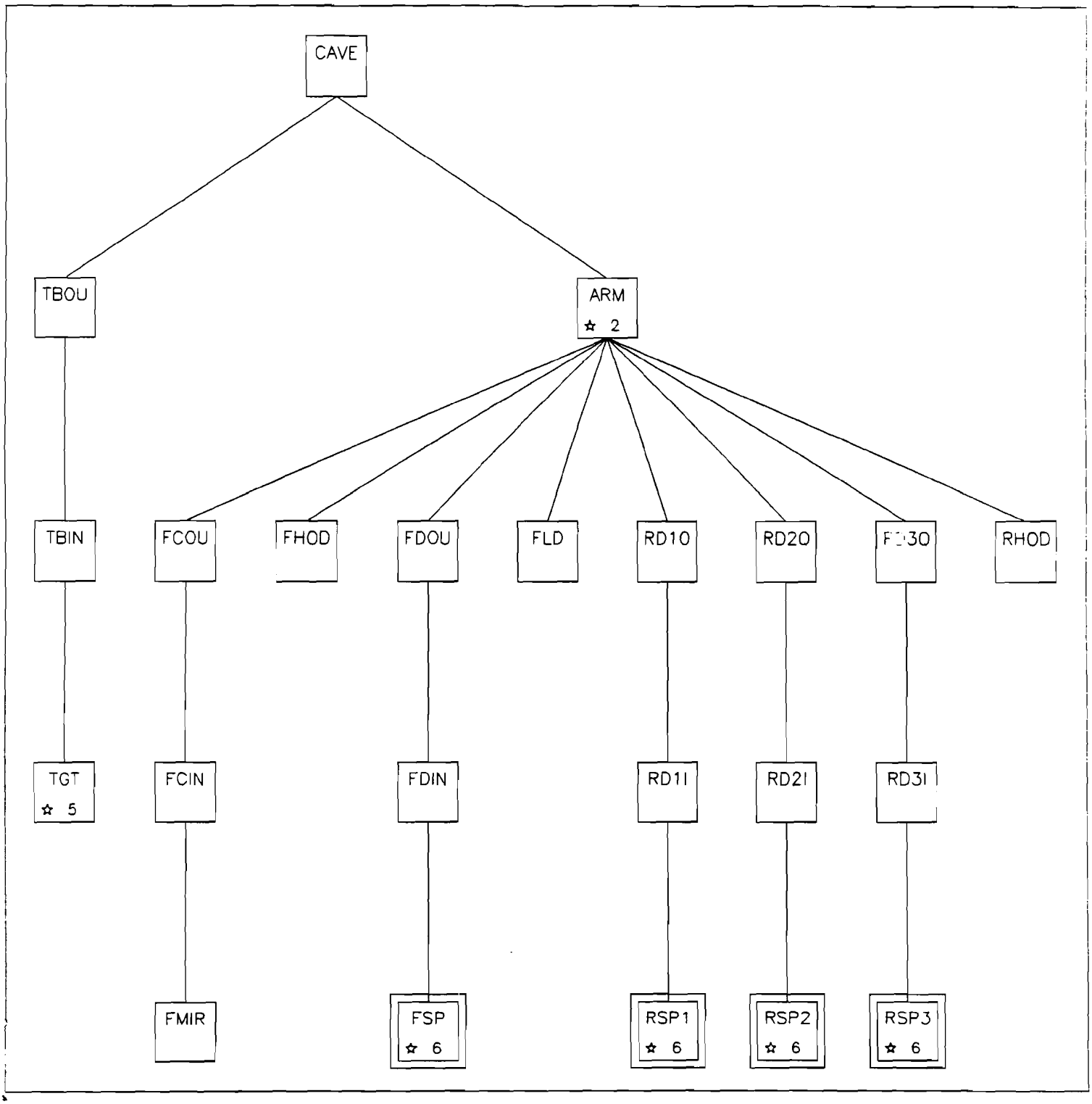











CAVE
TBOU
ARM
☆ 2
TBIN
FCOU
FHOD
FDOU
FLD
RD10
RD20
RHOD
TGT
☆ 5
FCIN
FDIN
RD11
RD21
RD31
FMIR
FSP
☆ 6
RSP1
☆ 6
RSP2
☆ 6
RSP3
☆ 6

```
*CMZ :  3.14/14 06/07/88  07.53.51  by  Rene Brun
*-- Author :
      SUBROUTINE UGEOM
C
C
*KEEP,GCLIST.
      COMMON/GCLIST/NHSTA,NGET ,NSAVE,NSETS,NPRIN,NGEOM,NVIEW,NPLOT
     +       ,NSTAT,LHSTA(20),LGET (20),LSAVE(20),LSETS(20),LPRIN(20)
     +             ,LGEOM(20),LVIEW(20),LPLOT(20),LSTAT(20)
C
*KEEP,GCONST.
      COMMON/GCONST/PI,TWOPI ,PIBY2,DEGRAD,RADDEG,CLIGHT ,BIG,EMASS
      COMMON/GCONSX/EMMU,PMASS,AVO
C
*KEND.
C
      COMMON/DLSFLD/ISWFLD,FLDVAL
C
      DIMENSION   CAVPAR(3),TGTPAR(3),TBIPAR(4),TBOPAR(4),ARMPAR(4)
     + ,FMIPAR(3),FCIPAR(5),FCOPAR(5),FHOPAR(3),FDIPAR(3),FDOPAR(3)
     + ,FLDPAR(3),RD1IPA(3),RD1OPA(3),RD2IPA(3),RD2OPA(3),RD3IPA(3)
     + ,RD3OPA(3),RHOPAR(3)
C
C      VOLUME SIZE PARAMETERS
C
      DATA CAVPAR/450.,100.,360./
      DATA TGTPAR/.47625,1.27,.0232/
      DATA TBIPAR/60.91172,5.07172,10.14,27.92/
      DATA TBOPAR/60.96,5.08,10.16,27.94/
      DATA ARMPAR/18.,210.,50.,190./
      DATA FMIPAR/36.83,12.7,.06/
      DATA FCIPAR/13.29941,50.69885,7.565299,13.90419,34.23/
      DATA FCOPAR/13.335,50.80,7.62,13.97,34.29/
      DATA FHOPAR/40.005,13.49375,.15875/
      DATA FDIPAR/41.91,13.97,4.2/
      DATA FDOPAR/41.917,13.977,4.207/
      DATA FLDPAR/55.88,19.05,25.40/
      DATA RD1IPA/70.000,25.500,7.800/
      DATA RD1OPA/70.007,25.507,7.807/
      DATA RD2IPA/85.000,29.500,7.800/
      DATA RD2OPA/85.007,29.507,7.807/
      DATA RD3IPA/100.000,33.500,7.800/
      DATA RD3OPA/100.007,33.507,7.807/
      DATA RHOPAR/150.5,47.,.3175/
C
C      LOCATIONS:
C          TGT VERSUS CAVE CENTER
C          FIELD AND DETECTORS VERSUS ARM CENTER
C          ANGLE SETTINGS OF LEFT AND RIGHT ARMS (DEG., ANGR IS NEG.)
C          LEFT AND RIGHT ARM VERSUS CAVE CENTER
C
      DATA ZTG/-100./
      DATA DFCO,DFHO,DFDO,DFLD,DRD1,DRD2,DRD3,DRHO/-154.77,-120.1625
     + ,-115.4,-84.92,-27.77,3.98,35.73,142.41/
      DATA ANGL,ANGR/45.,-45./
      DATA DARM/217./
      CANGL=COS(ANGL*DEGRAD)
      SANGL=SIN(ANGL*DEGRAD)
      CANGR=COS(ANGR*DEGRAD)
      SANGR=SIN(ANGR*DEGRAD)
      XLARM=DARM*SANGL
      ZLARM=DARM*CANGL+ZTG
      XRARM=DARM*SANGR
      ZRARM=DARM*CANGR+ZTG
C
C      CAVE REFERENCE FRAME:
```

```
C          OZ ALONG BEAM, OX HORIZONTAL, OY VERTICAL
C
      CALL GSVOLU('CAVE','BOX ',1,CAVPAR,3,ICAVE)
C
C      TARGET BOX SHIFTED UPSTREAM 100CM IN CAVE.
C           REFERENCE FRAME:
C               OZ ALONG BEAM, OX HORIZONTAL, OY VERTICAL.
C           CA DENSITY = 1.55 G/CM**3.
C           5 SEGMENTS.
C
      CALL GSVOLU('TGT ','BOX ',2,TGTPAR,3,ITGT )
      CALL GSVOLU('TBIN','TRD1',3,TBIPAR,4,ITBIN)
      CALL GSVOLU('TBOU','TRD1',4,TBOPAR,4,ITBOU)
C
C      LEFT AND RIGHT ARM REFERENCE FRAMES:
C           OZ ALONG CENTRAL RAY, OX HORIZONTAL, OY VERTICAL.
C
      CALL GSVOLU('ARM ','TRD1',1,ARMPAR,4,IARM)
      CALL GSVOLU('FMIR','BOX ',5,FMIPAR,3,IFMIR)
      CALL GSVOLU('FCIN','TRD2',6,FCIPAR,5,IFCIN)
      CALL GSVOLU('FCOU','TRD2',7,FCOPAR,5,IFCOU)
      CALL GSVOLU('FHOD','BOX ',8,FHOPAR,3,IFHOD)
      CALL GSVOLU('FDIN','BOX ',9,FDIPAR,3,IFDIN)
      CALL GSVOLU('FDOU','BOX ',4,FDOPAR,3,IFDOU)
      CALL GSVOLU('FLD ','BOX ',10,FLDPAR,3,IFLD )
      CALL GSVOLU('RD1I','BOX ',11,RD1IPA,3,IRD1I)
      CALL GSVOLU('RD1O','BOX ', 4,RD1OPA,3,IRD1O)
      CALL GSVOLU('RD2I','BOX ',11,RD2IPA,3,IRD2I)
      CALL GSVOLU('RD2O','BOX ', 4,RD2OPA,3,IRD2O)
      CALL GSVOLU('RD3I','BOX ',11,RD3IPA,3,IRD3I)
      CALL GSVOLU('RD3O','BOX ', 4,RD3OPA,3,IRD3O)
      CALL GSVOLU('RHOD','BOX ', 8,RHOPAR,3,IRHOD)
C
C         DRIFT CHAMBER SENSE PLANES
C
      FDIPAR(3)=.0025
      CALL GSVOLU('FSP ','BOX ',13,FDIPAR,3,IFSP )
      RD1IPA(3)=.0025
      CALL GSVOLU('RSP1','BOX ',14,RD1IPA,3,IRSP1)
      RD2IPA(3)=.0025
      CALL GSVOLU('RSP2','BOX ',14,RD2IPA,3,IRSP2)
      RD3IPA(3)=.0025
      CALL GSVOLU('RSP3','BOX ',14,RD3IPA,3,IRSP3)
C
C
      THLX=90.+ANGL
      THRX=90.+ANGR
      PHX=0.
      THY=90.
      PHY=90.
      THLZ=ANGL
      THRZ=-ANGR
      PHLZ=0.
      PHRZ=180.
      CALL GSROTM(1,THLX,PHX,THY,PHY,THLZ,PHLZ)
      CALL GSROTM(2,THRX,PHX,THY,PHY,THRZ,PHRZ)
C
C
      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')
C
```

```
      CALL GSPOS('ARM ',1,'CAVE',XLARM,0.,ZLARM,1,'ONLY')
      CALL GSPOS('ARM ',2,'CAVE',XRARM,0.,ZRARM,2,'ONLY')
C
      CALL GSPOS('FMIR',1,'FCIN',0.,0.,26.0350,0,'ONLY')
      CALL GSPOS('FCIN',1,'FCOU',0.,0., 0.    ,0,'ONLY')
      CALL GSPOS('FCOU',1,'ARM ',0.,0., DFCO  ,0,'ONLY')
C
      CALL GSPOS('FHOD',1,'ARM ',0.,0., DFHO  ,0,'ONLY')
C
      CALL GSPOS('FDIN',1,'FDOU',0.,0., 0.    ,0,'ONLY')
      CALL GSPOS('FDOU',1,'ARM ',0.,0., DFDO  ,0,'ONLY')
C
      CALL GSPOS('FLD ',1,'ARM ',0.,0., DFLD  ,0,'ONLY')
C
      CALL GSPOS('RD1I',1,'RD1O',0.,0., 0.    ,0,'ONLY')
      CALL GSPOS('RD1O',1,'ARM ',0.,0., DRD1  ,0,'ONLY')
C
      CALL GSPOS('RD2I',1,'RD2O',0.,0., 0.    ,0,'ONLY')
      CALL GSPOS('RD2O',1,'ARM ',0.,0., DRD2  ,0,'ONLY')
C
      CALL GSPOS('RD3I',1,'RD3O',0.,0., 0.    ,0,'ONLY')
      CALL GSPOS('RD3O',1,'ARM ',0.,0., DRD3  ,0,'ONLY')
C
      CALL GSPOS('RHOD',1,'ARM ',0.,0., DRHO  ,0,'ONLY')
C
      CALL GSPOS('FSP ',1,'FDIN',0.,0.,-2.9975,0,'ONLY')
      CALL GSPOS('FSP ',2,'FDIN',0.,0.,-1.7975,0,'ONLY')
      CALL GSPOS('FSP ',3,'FDIN',0.,0., -.5975,0,'ONLY')
      CALL GSPOS('FSP ',4,'FDIN',0.,0.,  .6025,0,'ONLY')
      CALL GSPOS('FSP ',5,'FDIN',0.,0., 1.8025,0,'ONLY')
      CALL GSPOS('FSP ',6,'FDIN',0.,0., 3.0025,0,'ONLY')
C
      CALL GSPOS('RSP1',1,'RD1I',0.,0.,-5.9975,0,'ONLY')
      CALL GSPOS('RSP1',2,'RD1I',0.,0.,-3.5975,0,'ONLY')
      CALL GSPOS('RSP1',3,'RD1I',0.,0.,-1.1975,0,'ONLY')
      CALL GSPOS('RSP1',4,'RD1I',0.,0., 1.2025,0,'ONLY')
      CALL GSPOS('RSP1',5,'RD1I',0.,0., 3.6025,0,'ONLY')
      CALL GSPOS('RSP1',6,'RD1I',0.,0., 6.0025,0,'ONLY')
C
      CALL GSPOS('RSP2',1,'RD2I',0.,0.,-5.9975,0,'ONLY')
      CALL GSPOS('RSP2',2,'RD2I',0.,0.,-3.5975,0,'ONLY')
      CALL GSPOS('RSP2',3,'RD2I',0.,0.,-1.1975,0,'ONLY')
      CALL GSPOS('RSP2',4,'RD2I',0.,0., 1.2025,0,'ONLY')
      CALL GSPOS('RSP2',5,'RD2I',0.,0., 3.6025,0,'ONLY')
      CALL GSPOS('RSP2',6,'RD2I',0.,0., 6.0025,0,'ONLY')
C
      CALL GSPOS('RSP3',1,'RD3I',0.,0.,-5.9975,0,'ONLY')
      CALL GSPOS('RSP3',2,'RD3I',0.,0.,-3.5975,0,'ONLY')
      CALL GSPOS('RSP3',3,'RD3I',0.,0.,-1.1975,0,'ONLY')
      CALL GSPOS('RSP3',4,'RD3I',0.,0., 1.2025,0,'ONLY')
      CALL GSPOS('RSP3',5,'RD3I',0.,0., 3.6025,0,'ONLY')
      CALL GSPOS('RSP3',6,'RD3I',0.,0., 6.0025,0,'ONLY')
C
C      CALL GSORD('ARM ',3)
      CALL GSORD('FDIN',3)
      CALL GSORD('RD1I',3)
      CALL GSORD('RD2I',3)
      CALL GSORD('RD3I',3)
C
      CALL GSTMED( 1,'AIR        $',15,0,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED( 2,'TARGET     $',17,0,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED( 3,'VACUUM     $',16,0,0,0., 0.,.0,.0,.001,.5,0,0)
      CALL GSTMED( 4,'MYLAR      $',18,0,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED( 5,'PYREX      $',19,0,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED( 6,'FREON13    $',20,0,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED( 7,'POLYETHYL.$',22,0,0,0.,10.,.2,.1,.001,.5,0,0)
```

```
      CALL GSTMED( 8,'SCINTILL. $',23,0,0,0.,10.,.2,.1,.01,.5,0,0)
      CALL GSTMED( 9,'FRONTDRIFT$',24,0,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED(10,'AIR+FIELD $',15,0,ISWFLD,6.,10.,.2,.1,.01,.5,0,0)
      CALL GSTMED(11,'REAR DRIFT$',25,0,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED(12,'CO2       $',21,0,0,0.,10.,.2,.1,.01,.5,0,0)
      CALL GSTMED(13,'FR.DR.S.P.$',24,1,0,0.,10.,.2,.1,.001,.5,0,0)
      CALL GSTMED(14,'RE.DR.S.P.$',25,1,0,0.,10.,.2,.1,.001,.5,0,0)
C
C
      IF(IUCOMP(4HVOLU,LPRIN,NPRIN).NE.0)CALL GPVOLU(0)
      IF(IUCOMP(4HROTM,LPRIN,NPRIN).NE.0)CALL GPROTM(0)
      IF(IUCOMP(4HTMED,LPRIN,NPRIN).NE.0)CALL GPTMED(0)
      IF(IUCOMP(4HMATE,LPRIN,NPRIN).NE.0)CALL GPMATE(0)
      IF(IUCOMP(4HPART,LPRIN,NPRIN).NE.0)CALL GPPART(0)
C
C
      CALL GGCLOS
C
      RETURN
      END
```

@9900#9900,'Z68-300.89.06','2.0','GEANT-SET','1.0','CERN','','',''
#9901,1991,11,12,14,01,09
@9901,1,'DLS '#9905,3,1.
,1,0.001,0,-300.,300.,-300.,300.,-300.,300.
#9920,:12,0.005,:14,0,:15,0
@301,2#301,1.,0.,0.,0.,1.,0.,0.,0.,1.,0.,0.,0.
@50,3,:5,2#60,0.9524999856948853,2.5399999618530273,0.04639999994099140@302,4#317
,-0.4762500,-1.2700000,-0.0232000@100,5,:5,2,:9,'TGT'#101,!3,!4@50,6,:5,2#60,0.9
524999856948853,2.5399999618530273,0.04639999994099140@302,7#317,-0.4762500,-1.27
00000,-0.0232000@100,8,:5,2,:9,'TGT'#101,!6,!7@50,9,:5,2#60,0.9524999856948853,2
.5399999618530273,0.04639999994099140@302,10#317,-0.4762500,-1.2700000,-0.0232000
@100,11,:5,2,:9,'TGT'#101,!9,!10@50,12,:5,2#60,0.9524999856948853,2.539999961853
0273,0.04639999994099140@302,13#317,-0.4762500,-1.2700000,-0.0232000@100,14,:5,2,
:9,'TGT'#101,!12,!13@50,15,:5,2#60,0.9524999856948853,2.5399999618530273,0.04639
99994099140@302,16#317,-0.4762500,-1.2700000,-0.0232000@100,17,:5,2,:9,'TGT'#101
,!15,!16@50,18,:5,2#60,80.0100021362304690,26.9874992370605470,0.317499995231628
4@302,19#317,-40.0050011,-13.4937496,-0.1587500@100,20,:5,2,:9,'FHOD'#101,!18,!1
9@50,21,:5,2#60,111.7600021362304700,38.0999984741210940,50.7999992370605470@302
,22#317,-55.8800011,-19.0499992,-25.3999996@100,23,:5,2,:9,'FLD'#101,!21,!22@50,
24,:5,2#60,301.0000000000000000,94.0000000000000000,0.6349999904632568@302,25#31
7,-150.5000000,-47.0000000,-0.3175000@100,26,:5,2,:9,'RHOD'#101,!24,!25@50,27,:5
,2#60,73.6600036621093750,25.3999996185302730,0.11999999731779 10@302,28#317,-36.
8300018,-12.6999998,-0.0600000@100,29,:5,2,:9,'FMIR'#101,!27,!28@50,30,:5,2#60,8
3.8199996948242190,27.9400005340576170,0.00499999998882413@302,31#317,-41.9099998
,-13.9700003,-0.0025000@100,32,:5,2,:9,'FSP'#101,!30,!31@50,33,:5,2#60,83.819999
6948242190,27.9400005340576170,0.00499999998882413@302,34#317,-41.9099998,-13.970
0003,-0.0025000@100,35,:5,2,:9,'FSP'#101,!33,!34@50,36,:5,2#60,83.81999969482421
90,27.9400005340576170,0.00499999998882413@302,37#317,-41.9099998,-13.9700003,-0.
0025000@100,38,:5,2,:9,'FSP'#101,!36,!37@50,39,:5,2#60,83.8199996948242190,27.94
00005340576170,0.00499999998882413@302,40#317,-41.9099998,-13.9700003,-0.0025000@
100,41,:5,2,:9,'FSP'#101,!39,!40@50,42,:5,2#60,83.8199996948242190,27.9400005340
576170,0.00499999998882413@302,43#317,-41.9099998,-13.9700003,-0.0025000@100,44,:
5,2,:9,'FSP'#101,!42,!43@50,45,:5,2#60,83.8199996948242190,27.9400005340576170,0
.00499999998882413@302,46#317,-41.9099998,-13.9700003,-0.0025000@100,47,:5,2,:9,'
FSP'#101,!45,!46@50,48,:5,2#60,140.0000000000000000,51.0000000000000000,0.004999
9998882413@302,49#317,-70.0000000,-25.5000000,-0.0025000@100,50,:5,2,:9,'RSP1'#1
01,!48,!49@50,51,:5,2#60,140.0000000000000000,51.0000000000000000,0.004999999888
2413@302,52#317,-70.0000000,-25.5000000,-0.0025000@100,53,:5,2,:9,'RSP1'#101,!51
,!52@50,54,:5,2#60,140.0000000000000000,51.0000000000000000,0.00499999998882413@3
02,55#317,-70.0000000,-25.5000000,-0.0025000@100,56,:5,2,:9,'RSP1'#101,!54,!55@5
0,57,:5,2#60,140.0000000000000000,51.0000000000000000,0.00499999998882413@302,58#
317,-70.0000000,-25.5000000,-0.0025000@100,59,:5,2,:9,'RSP1'#101,!57,!58@50,60,:
5,2#60,140.0000000000000000,51.0000000000000000,0.00499999998882413@302,61#317,-7
0.0000000,-25.5000000,-0.0025000@100,62,:5,2,:9,'RSP1'#101,!60,!61@50,63,:5,2#60
,140.0000000000000000,51.0000000000000000,0.00499999998882413@302,64#317,-70.0000
000,-25.5000000,-0.0025000@100,65,:5,2,:9,'RSP1'#101,!63,!64@50,66,:5,2#60,170.0
000000000000000,59.0000000000000000,0.00499999998882413@302,67#317,-85.0000000,-2
9.5000000,-0.0025000@100,68,:5,2,:9,'RSP2'#101,!66,!67@50,69,:5,2#60,170.0000000
000000000,59.0000000000000000,0.00499999998882413@302,70#317,-85.0000000,-29.5000
000,-0.0025000@100,71,:5,2,:9,'RSP2'#101,!69,!70@50,72,:5,2#60,170.0000000000000
000,59.0000000000000000,0.00499999998882413@302,73#317,-85.0000000,-29.5000000,-0
.0025000@100,74,:5,2,:9,'RSP2'#101,!72,!73@50,75,:5,2#60,170.0000000000000000,59
.0000000000000000,0.00499999998882413@302,76#317,-85.0000000,-29.5000000,-0.00250
00@100,77,:5,2,:9,'RSP2'#101,!75,!76@50,78,:5,2#60,170.0000000000000000,59.00000
00000000000,0.00499999998882413@302,79#317,-85.0000000,-29.5000000,-0.0025000@100
,80,:5,2,:9,'RSP2'#101,!78,!79@50,81,:5,2#60,170.0000000000000000,59.00000000000
00000,0.00499999998882413@302,82#317,-85.0000000,-29.5000000,-0.0025000@100,83,:5
,2,:9,'RSP2'#101,!81,!82@50,84,:5,2#60,200.0000000000000000,67.0000000000000000,
0.00499999998882413@302,85#317,-100.0000000,-33.5000000,-0.0025000@100,86,:5,2,:9
,'RSP3'#101,!84,!85@50,87,:5,2#60,200.0000000000000000,67.0000000000000000,0.004
99999998882413@302,88#317,-100.0000000,-33.5000000,-0.0025000@100,89,:5,2,:9,'RSP
3'#101,!87,!88@50,90,:5,2#60,200.0000000000000000,67.0000000000000000,0.00499999
98882413@302,91#317,-100.0000000,-33.5000000,-0.0025000@100,92,:5,2,:9,'RSP3'#10
1,!90,!91@50,93,:5,2#60,200.0000000000000000,67.0000000000000000,0.0049999998882
413@302,94#317,-100.0000000,-33.5000000,-0.0025000@100,95,:5,2,:9,'RSP3'#101,!93
,!94@50,96,:5,2#60,200.0000000000000000,67.0000000000000000,0.00499999988882413@3

02,97#317,-100.0000000,-33.5000000,-0.0025000@100,98,:5,2,:9,'RSP3'#101,!96,!97@
50,99,:5,2#60,200.0000000000000000,67.0000000000000000,0.0049999998882413@302,10
0#317,-100.0000000,-33.5000000,-0.0025000@100,101,:5,2,:9,'RSP3'#101,!99,!100@30
2,102#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,
0.0000000,1.0000000,0.0000000,0.0000000,-5.0799999@100,103,:5,2,:9,'TGT'#101,!17
,!102@302,104#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.
0000000,0.0000000,1.0000000,0.0000000,0.0000000,-2.5400000@100,105,:5,2,:9,'TGT'
#101,!17,!104@302,106#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.00
00000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000@100,107,:5,2,:
9,'TGT'#101,!17,!106@302,108#301,1.0000000,0.0000000,0.0000000,0.0000000,1.00000
00,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,2.5400000@100,109
,:5,2,:9,'TGT'#101,!17,!108@302,110#301,1.0000000,0.0000000,0.0000000,0.0000000,
1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,5.0799999@
100,111,:5,2,:9,'TGT'#101,!17,!110@103,112,:5,2#3,3,2,-60.9117203,-10.1400003,-2
7.9200001,60.9117203,-10.1400003,-27.9200001,60.9117203,10.1400003,-27.9200001,-
60.9117203,10.1400003,-27.9200001@103,113,:5,2#3,3,2,-5.0717201,-10.1400003,27.9
200001,5.0717201,-10.1400003,27.9200001,5.0717201,10.1400003,27.9200001,-5.07172
01,10.1400003,27.9200001@100,114,:5,2,:9,'TBIN'#145,!112,!113@100,115,:5,2#100,3
,!114,!103,!105,!107,!109,!111@510,116,:5,2#510,1,0,0,0,!115,!103,!105,!107,!109
,!111@302,117#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.
0000000,0.0000000,1.0000000,0.0000000,0.0000000,26.0349998@100,118,:5,2,:9,'FMIR
'#101,!29,!117@103,119,:5,2#3,3,2,-13.2994099,-7.5652990,-34.2299995,13.2994099,
-7.5652990,-34.2299995,13.2994099,7.5652990,-34.2299995,-13.2994099,7.5652990,-3
4.2299995@103,120,:5,2#3,3,2,-50.6988487,-13.9041901,34.2299995,50.6988487,-13.9
041901,34.2299995,50.6988487,13.9041901,34.2299995,-50.6988487,13.9041901,34.229
9995@100,121,:5,2,:9,'FCIN'#145,!119,!120@100,122,:5,2#100,3,!121,!118@510,123,:
5,2#510,1,0,0,0,!122,!118@302,124#301,1.0000000,0.0000000,0.0000000,0.0000000,1.
0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-2.9974999@1
00,125,:5,2,:9,'FSP'#101,!47,!124@302,126#301,1.0000000,0.0000000,0.0000000,0.00
00000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-1.7
975000@100,127,:5,2,:9,'FSP'#101,!47,!126@302,128#301,1.0000000,0.0000000,0.0000
000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000
000,-0.5975000@100,129,:5,2,:9,'FSP'#101,!47,!128@302,130#301,1.0000000,0.000000
0,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.000000
0,0.0000000,0.6025000@100,131,:5,2,:9,'FSP'#101,!47,!130@302,132#301,1.0000000,0
.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0
.0000000,0.0000000,1.8025000@100,133,:5,2,:9,'FSP'#101,!47,!132@302,134#301,1.00
00000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.00
00000,0.0000000,0.0000000,3.0025001@100,135,:5,2,:9,'FSP'#101,!47,!134@50,136,:5
,2#60,83.8199996948242190,27.9400005340576170,8.3999996185302734@302,137#317,-41
.9099998,-13.9700003,-4.1999998@100,138,:5,2,:9,'FDIN'#101,!136,!137@100,139,:5,
2#100,3,!138,!125,!127,!129,!131,!133,!135@510,140,:5,2#510,1,0,0,0,!139,!125,!1
27,!129,!131,!133,!135@302,141#301,1.0000000,0.0000000,0.0000000,0.0000000,1.000
0000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-5.9974999@100,
142,:5,2,:9,'RSP1'#101,!65,!141@302,143#301,1.0000000,0.0000000,0.0000000,0.0000
000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-3.597
5001@100,144,:5,2,:9,'RSP1'#101,!65,!143@302,145#301,1.0000000,0.0000000,0.00000
00,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.00000
00,-1.1975000@100,146,:5,2,:9,'RSP1'#101,!65,!145@302,147#301,1.0000000,0.000000
0,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.000000
0,0.0000000,1.2025000@100,148,:5,2,:9,'RSP1'#101,!65,!147@302,149#301,1.0000000,
0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,
0.0000000,0.0000000,3.6025000@100,150,:5,2,:9,'RSP1'#101,!65,!149@302,151#301,1.
0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.
0000000,0.0000000,0.0000000,6.0025001@100,152,:5,2,:9,'RSP1'#101,!65,!151@50,153
,:5,2#60,140.0000000000000000,51.0000000000000000,15.6000003814697270@302,154#31
7,-70.0000000,-25.5000000,-7.8000002@100,155,:5,2,:9,'RD1I'#101,!153,!154@100,15
6,:5,2#100,3,!155,!142,!144,!146,!148,!150,!152@510,157,:5,2#510,1,0,0,0,!156,!1
42,!144,!146,!148,!150,!152@302,158#301,1.0000000,0.0000000,0.0000000,0.0000000,
1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-5.9974999
@100,159,:5,2,:9,'RSP2'#101,!83,!158@302,160#301,1.0000000,0.0000000,0.0000000,0
.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-
3.5975001@100,161,:5,2,:9,'RSP2'#101,!83,!160@302,162#301,1.0000000,0.0000000,0.
0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.
0000000,-1.1975000@100,163,:5,2,:9,'RSP2'#101,!83,!162@302,164#301,1.0000000,0.0
000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0

000000,0.0000000,1.2025000@100,165,:5,2,:9,'RSP2'#101,!83,!164@302,166#301,1.000
0000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.000
0000,0.0000000,0.0000000,3.6025000@100,167,:5,2,:9,'RSP2'#101,!83,!166@302,168#3
01,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.00000
00,1.0000000,0.0000000,0.0000000,6.0025001@100,169,:5,2,:9,'RSP2'#101,!83,!168@5
0,170,:5,2#60,170.0000000000000000,59.0000000000000000,15.6000003814697270@302,1
71#317,-85.0000000,-29.5000000,-7.8000002@100,172,:5,2,:9,'RD2I'#101,!170,!171@1
00,173,:5,2#100,3,!172,!159,!161,!163,!165,!167,!169@510,174,:5,2#510,1,0,0,0,!1
73,!159,!161,!163,!165,!167,!169@302,175#301,1.0000000,0.0000000,0.0000000,0.000
0000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-5.99
74999@100,176,:5,2,:9,'RSP3'#101,!101,!175@302,177#301,1.0000000,0.0000000,0.000
0000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.000
0000,-3.5975001@100,178,:5,2,:9,'RSP3'#101,!101,!177@302,179#301,1.0000000,0.000
0000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.000
0000,0.0000000,-1.1975000@100,180,:5,2,:9,'RSP3'#101,!101,!179@302,181#301,1.000
0000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.000
0000,0.0000000,0.0000000,1.2025000@100,182,:5,2,:9,'RSP3'#101,!101,!181@302,183#
301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000
000,1.0000000,0.0000000,0.0000000,3.6025000@100,184,:5,2,:9,'RSP3'#101,!101,!183
@302,185#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.00000
00,0.0000000,1.0000000,0.0000000,0.0000000,6.0025001@100,186,:5,2,:9,'RSP3'#101,
!101,!185@50,187,:5,2#60,200.0000000000000000,67.0000000000000000,15.60000038146
97270@302,188#317,-100.0000000,-33.5000000,-7.8000002@100,189,:5,2,:9,'RD3I'#101
,!187,!188@100,190,:5,2#100,3,!189,!176,!178,!180,!182,!184,!186@510,191,:5,2#51
0,1,0,0,0,!190,!176,!178,!180,!182,!184,!186@302,192#301,1.0000000,0.0000000,0.0
000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0
000000,0.0000000@100,193,:5,2,:9,'TBIN'#101,!116,!192@103,194,:5,2#3,3,2,-60.959
9991,-10.1599998,-27.9400005,60.9599991,-10.1599998,-27.9400005,60.9599991,10.15
99998,-27.9400005,-60.9599991,10.1599998,-27.9400005@103,195,:5,2#3,3,2,-5.07999
99,-10.1599998,27.9400005,5.0799999,-10.1599998,27.9400005,5.0799999,10.1599998,
27.9400005,-5.0799999,10.1599998,27.9400005@100,196,:5,2,:9,'TBOU'#145,!194,!195
@100,197,:5,2#100,3,!196,!193@510,198,:5,2#510,1,0,0,0,!197,!193@302,199#301,1.0
000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0
000000,0.0000000,0.0000000,0.0000000@100,200,:5,2,:9,'FCIN'#101,!123,!199@103,20
1,:5,2#3,3,2,-13.3350000,-7.6199999,-34.2900009,13.3350000,-7.6199999,-34.290000
9,13.3350000,7.6199999,-34.2900009,-13.3350000,7.6199999,-34.2900009@103,202,:5,
2#3,3,2,-50.7999992,-13.9700003,34.2900009,50.7999992,-13.9700003,34.2900009,50.
7999992,13.9700003,34.2900009,-50.7999992,13.9700003,34.2900009@100,203,:5,2,:9,
'FCOU'#145,!201,!202@100,204,:5,2#100,3,!203,!200@510,205,:5,2#510,1,0,0,0,!204,
!200@302,206#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0
000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000@100,207,:5,2,:9,'FDIN'#
101,!140,!206@50,208,:5,2#60,83.8339996337890630,27.9540004730224610,8.413999557
4951172@302,209#317,-41.9169998,-13.9770002,-4.2069998@100,210,:5,2,:9,'FDOU'#10
1,!208,!209@100,211,:5,2#100,3,!210,!207@510,212,:5,2#510,1,0,0,0,!211,!207@302,
213#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.
0000000,1.0000000,0.0000000,0.0000000,0.0000000@100,214,:5,2,:9,'RD1I'#101,!157,
!213@50,215,:5,2#60,140.0140075683593700,51.0139999389648440,15.6140003204345700
@302,216#317,-70.0070038,-25.5070000,-7.8070002@100,217,:5,2,:9,'RD1O'#101,!215,
!216@100,218,:5,2#100,3,!217,!214@510,219,:5,2#510,1,0,0,0,!218,!214@302,220#301
,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000
,1.0000000,0.0000000,0.0000000,0.0000000@100,221,:5,2,:9,'RD2I'#101,!174,!220@50
,222,:5,2#60,170.0140075683593700,59.0139999389648440,15.6140003204345700@302,22
3#317,-85.0070038,-29.5070000,-7.8070002@100,224,:5,2,:9,'RD2O'#101,!222,!223@10
0,225,:5,2#100,3,!224,!221@510,226,:5,2#510,1,0,0,0,!225,!221@302,227#301,1.0000
000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000
000,0.0000000,0.0000000,0.0000000@100,228,:5,2,:9,'RD3I'#101,!191,!227@50,229,:5
,2#60,200.0140075683593700,67.0139999389648440,15.6140003204345700@302,230#317,-
100.0070038,-33.5070000,-7.8070002@100,231,:5,2,:9,'RD3O'#101,!229,!230@100,232,
:5,2#100,3,!231,!228@510,233,:5,2#510,1,0,0,0,!232,!228@302,234#301,1.0000000,0.
0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.
0000000,0.0000000,-154.7700043@100,235,:5,2,:9,'FCOU'#101,!205,!234@302,236#301,
1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,
1.0000000,0.0000000,0.0000000,-120.1624985@100,237,:5,2,:9,'FHOD'#101,!20,!236@3
02,238#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000
,0.0000000,1.0000000,0.0000000,0.0000000,-115.4000015@100,239,:5,2,:9,'FDOU'#101
,!212,!238@302,240#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.00000

00,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-84.9199982@100,241,:5,2,:9
,'FLD'#101,!23,!240@302,242#301,1.0000000,0.0000000,0.0000000,0.0000000,1.000000
0,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,-27.7700005@100,24
3,:5,2,:9,'RD1O'#101,!219,!242@302,244#301,1.0000000,0.0000000,0.0000000,0.00000
00,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,3.98000
00@100,245,:5,2,:9,'RD2O'#101,!226,!244@302,246#301,1.0000000,0.0000000,0.000000
0,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.000000
0,35.7299995@100,247,:5,2,:9,'RD3O'#101,!233,!246@302,248#301,1.0000000,0.000000
0,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.000000
0,0.0000000,142.4100037@100,249,:5,2,:9,'RHOD'#101,!26,!248@103,250,:5,2#3,3,2,-
18.0000000,-50.0000000,-190.0000000,18.0000000,-50.0000000,-190.0000000,18.00000
00,50.0000000,-190.0000000,-18.0000000,50.0000000,-190.0000000@103,251,:5,2#3,3,
2,-210.0000000,-50.0000000,190.0000000,210.0000000,-50.0000000,190.0000000,210.0
000000,50.0000000,190.0000000,-210.0000000,50.0000000,190.0000000@100,252,:5,2,:
9,'ARM'#145,!250,!251@100,253,:5,2#100,3,!252,!235,!237,!239,!241,!243,!245,!247
,!249@510,254,:5,2#510,1,0,0,0,!253,!235,!237,!239,!241,!243,!245,!247,!249@302,
255#301,1.0000000,0.0000000,0.0000000,0.0000000,1.0000000,0.0000000,0.0000000,0.
0000000,1.0000000,0.0000000,0.0000000,-100.0000000@100,256,:5,2,:9,'TBOU'#101,!1
98,!255@302,257#301,0.7071068,0.0000000,-0.7071068,0.0000000,1.0000000,0.0000000
,0.7071068,0.0000000,0.7071068,153.4421844,0.0000000,53.4421692@100,258,:5,2,:9,
'ARM'#101,!254,!257@302,259#301,0.7071068,0.0000000,0.7071068,0.0000000,1.000000
0,0.0000000,-0.7071068,-0.0000001,0.7071068,-153.4421844,0.0000000,53.4421692@10
0,260,:5,2,:9,'ARM'#101,!254,!259@50,261,:5,2#60,900.0000000000000000,200.000000
0000000000,720.0000000000000000@302,262#317,-450.0000000,-100.0000000,-360.00000
00@100,263,:5,2,:9,'CAVE'#101,!261,!262@100,264,:5,2#100,3,!263,!256,!258,!260@5
10,265,:9,'CAVE'#510,1,0,0,0,!264,!256,!258,!260@9998,266@9999,268