Solar Neutron Telescope Gornergrat
Monte Carlo Simulation of Detector Properties

Diplomarbeit

der Philosophisch-naturwissenschaftlichen Fakultät

der Universität Bern

vorgelegt von

Michael R. Moser
von Biglen BE

2002

Leiter der Arbeit:

Prof. Dr. Erwin O. Flückiger
Physikalisches Institut

Contents

Chapter 1

Introduction

In 1951 Biermann, Haxel, and Schlüter postulated that during high energetic solar eruptions (*solar flares*), such as illustrated in Figure 1.1, neutrons may be produced with energies high enough to reach the Earth before they decay. In contrast to the charged particles accelerated and emitted in association with a flare or a *coronal mass ejection (CME)*, neutrons are not influenced by the magnetic fields at the Sun, in the interplanetary space, and near Earth. Therefore, the observation of solar neutrons may provide direct information on the acceleration processes during high energetic solar eruptions.

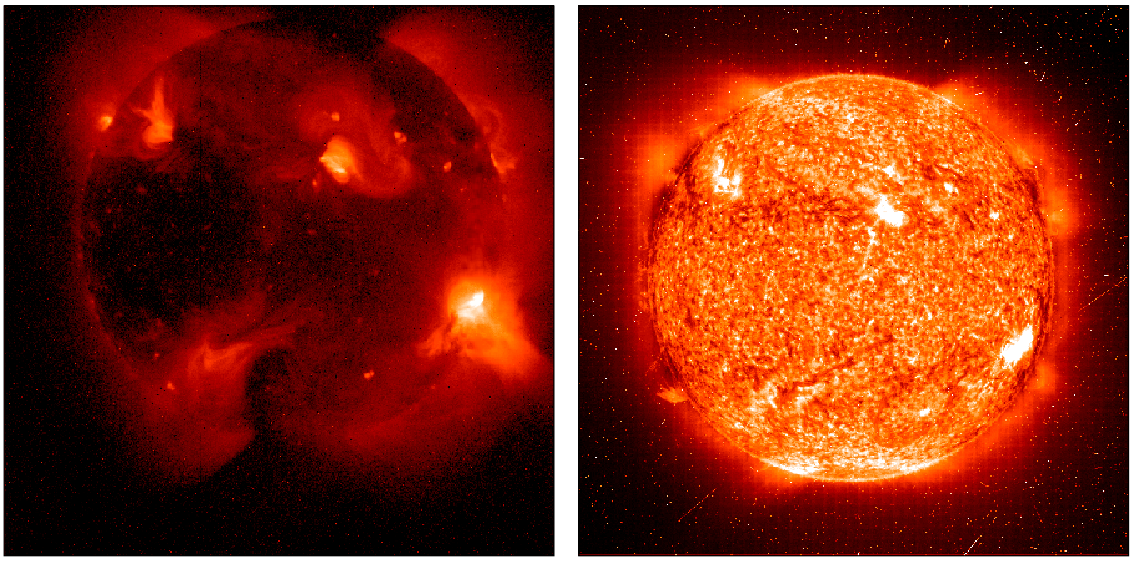


Figure 1.1: The Sun after a class X9.4 flare on November 6, 1997. Associated solar neutrons were detected at Earth (Tsuchiya et al., 2001). The image on the left corresponds to X-rays in the energy range 0.25–4.0 keV as observed by the SXT instrument on the Yohkoh spacecraft (from `http://umbra.nascom.nasa.gov/`) and the image on the right corresponds to a wavelength of 304 Å as observed by the EIT instrument on SOHO (from `http://sohowww.nascom.nasa.gov/`).

Evidence for the existence of neutrons at the Sun, producing the n-p-capture γ-ray line of 2.223 MeV, has been reported e.g. by Share et al. (1982). On June 21, 1980, the *Gamma Ray Spectrometer (GRS)* aboard the *Solar Maximum Mission (SMM)* satellite detected solar neutrons near Earth (Chupp et al., 1982). The first identification of solar neutrons

at Earth took place two years later, on June 3, 1982, by neutron monitor measurements at Jungfraujoch, Lomnicky Stit, and Rome (Debrunner et al., 1983; Chupp et al., 1987). This discovery initiated extensive theoretical and experimental work on the production of solar neutrons during solar flares, their propagation, and detection near and at the Earth. *COMPTEL*, the Compton telescope on the *Compton Gamma Ray Observatory (CGRO)* was built to detect γ-rays and solar neutrons in space (Ryan et al., 1993). At favorable observational locations at Earth, such as Haleakala, Hawaii (Pyle and Simpson, 1991), standardized neutron monitors were set up. Additionally, new ground-based detectors with enhanced sensitivity to solar neutrons were developed (Shibata et al., 1991; Muraki et al., 1993).

Initiated by the *Solar-Terrestrial Environment Laboratory (STEL)* of the Nagoya University in Japan, a new global network of *Solar Neutron Telescopes* was set up. Due to the longitude distribution of the locations it is possible to observe solar neutrons during 24 hours a day. Figure 1.2 shows the STEL network and Table 1.1 lists the geographical locations of the detectors which are in operation today.

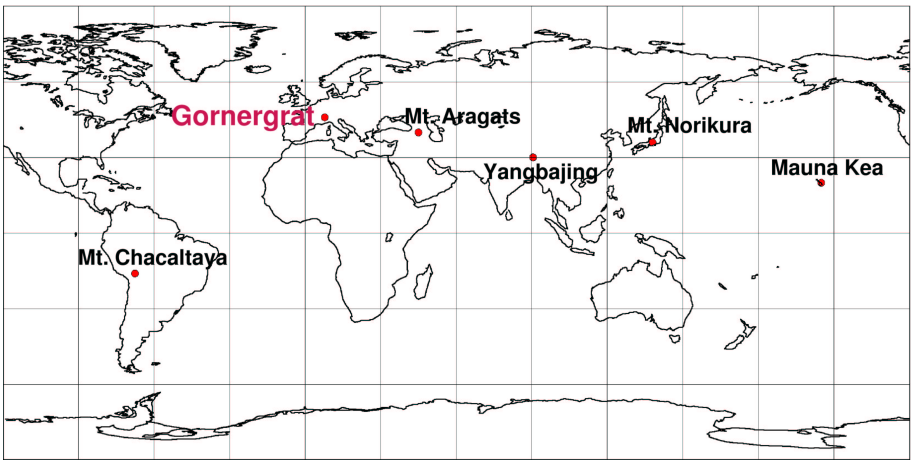


Figure 1.2: The solar neutron telescope network of STEL.

Location	Altitude	Longitude	Latitude
Gornergrat, Switzerland	3135 m a.s.l.	7.78°E	45.98°N
Mauna Kea, USA	4200 m a.s.l.	155°W	20°N
Mt. Aragats, Armenia	3200 m a.s.l.	45°E	40°N
Mt. Chacaltaya, Bolivia	5250 m a.s.l.	68°W	16°S
Mt. Norikura, Japan	2770 m a.s.l.	138°E	36°N
Yangbajing, Tibet	4300 m a.s.l.	90°E	30°N

Table 1.1: Geographical locations of the STEL solar neutron telescopes.

The Swiss solar neutron telescope (SONTEL), the European cornerstone of this network, was installed in January 1998 at Gornergrat. The detector housing at the Belvedere platform of Gornergrat Kulm is shown in Figure 1.3. Figure 1.4 shows the inside of the laboratory container with the two data-taking computers and the electronics rack in front of the proportional counters of the detector.

Figure 1.3: Detector housing of SONTEL at Gornergrat.

Figure 1.4: The SONTEL laboratory container at Gornergrat with the two data-taking computers (left), the electronics rack (right), and the proportional counter tubes (back).

In order to interpret the SONTEL recordings during solar neutron events, it is necessary to know the relationship between the counting rates and the primary particle flux penetrating the Earth's atmosphere. This relationship can hardly be determined experimentally. Also, in particle beam calibrations the atmospheric cascade of primary cosmic ray particles cannot realistically be reproduced. An analytical treatment is too difficult since the complex properties of the atmosphere and of the detector have to be considered.

Within the scope of this work, we therefore developed a Monte Carlo application to determine the detector properties of SONTEL at Gornergrat theoretically. The Monte Carlo code is based on CERN's GEANT3 libraries which allow to simulate the interaction processes of radiation with matter. From this application we first determined the efficiencies of the various SONTEL channels. The same code also allowed to simulate the average counting rates. As input we used secondary cosmic ray spectra from literature adapted to Gornergrat altitude. For the simulation of the detector response to *Ground Level Enhancements (GLE)*, we developed another GEANT3 application to determine the corresponding secondary particle spectra above the detector. We simulated the detector response for two historic solar particle events, i.e. the solar neutron event of June 3, 1982, and the solar proton event of April 15, 2001. The latter has been recorded by SONTEL at Gornergrat, providing the possibility for a verification of the theoretical results.

In the following, we first describe the detector and its technical details. We then give a summary of the secondary cosmic ray spectra adapted to Gornergrat altitude. The Monte Carlo applications are described in Chapter 4, and the obtained simulation results are presented in detail in Chapter 5. The last two chapters contain an extended discussion of the results as well as a comparison to experimental data. An appendix finally lists the source code of all the programs.

Chapter 2

Solar Neutron Telescope Gornergrat

The *Solar Neutron Telescope (SONTEL)* described in this thesis was installed at Gornergrat in January 1998 as the European cornerstone of the worldwide network of solar neutron detectors initiated by the *Solar-Terrestrial Environment Laboratory (STEL)* of the Nagoya University, Nagoya, Japan (Flückiger et al., 1998). It was upgraded in autumn 1999 (Bütikofer et al., 2001). Figure 2.1 schematically shows the detector in its present configuration.

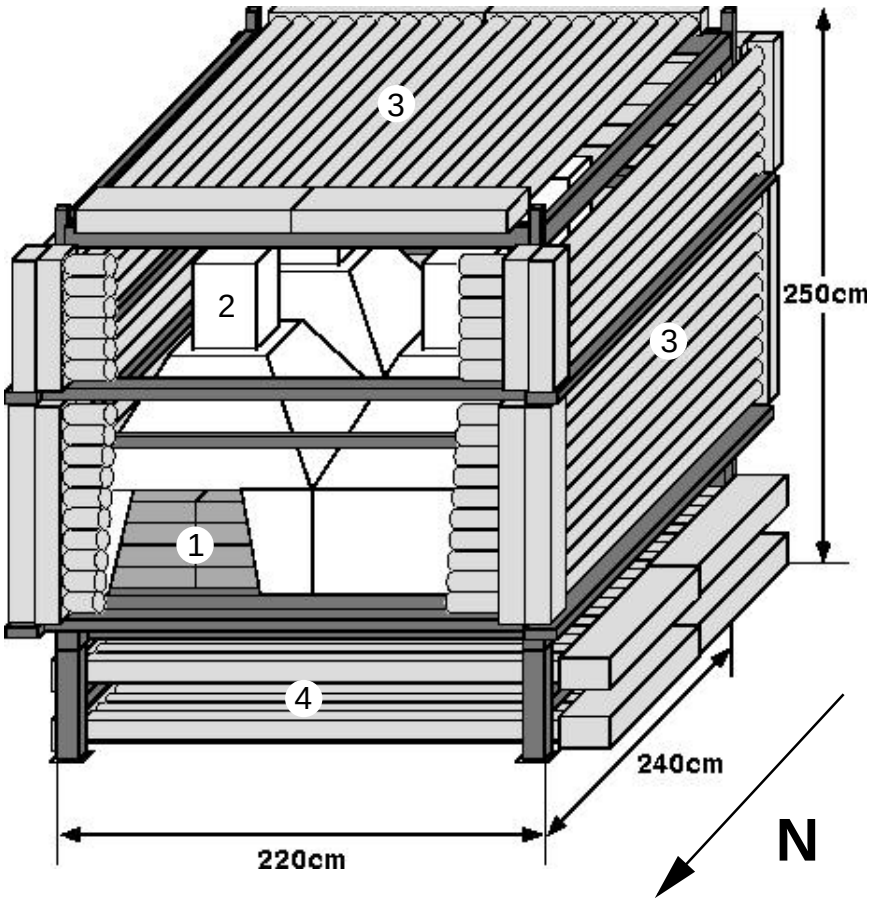


Figure 2.1: Schematic view of SONTEL at Gornergrat. For details see text.

In the following paragraphs the principles of operation of the detector will be described and the most important technical details summarized.

2.1 Experimental Setup

The *Solar Neutron Telescope* at Gornergrat consists of a horizontal matrix of four 1 m × 1 m × 0.4 m plastic scintillator units (see 1 in Figure 2.1). Incident neutrons may produce protons by nuclear interaction processes within the scintillator material. These protons can then be detected.

The scintillator consists mainly of hydrogen and carbon (polystyrene). Recoil protons are then produced by two interaction processes. The first interaction process is elastic scattering of the neutrons at hydrogen nuclei

$$\mathrm{n} + \mathrm{p} \rightarrow \mathrm{p} + \mathrm{n}. \tag{2.1}$$

The second interaction process is inelastic scattering on carbon nuclei

$$\mathrm{n} + \mathrm{C} \rightarrow \mathrm{p} + \mathrm{X}. \tag{2.2}$$

The cross sections of the two interaction processes are illustrated in Figure 2.2. Inelastic processes are negligible below about 5 MeV but become dominant at neutron energies higher than 20 MeV.

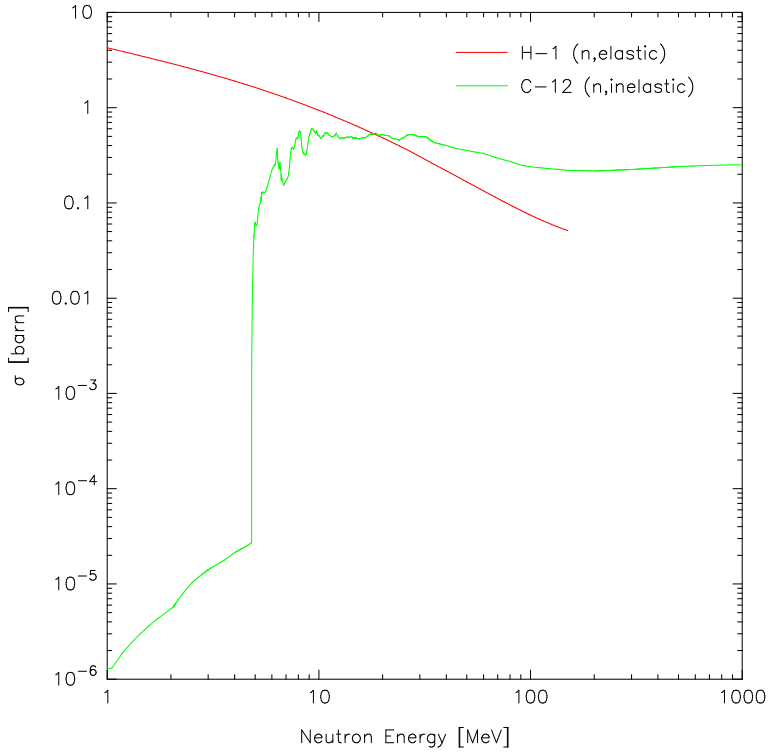


Figure 2.2: Interaction cross sections of the processes $\mathrm{n} + \mathrm{p} \rightarrow \mathrm{p} + \mathrm{n}$ and $\mathrm{n} + \mathrm{C} \rightarrow \mathrm{p} + \mathrm{X}$ (from the official ENDF database available at `http://www-nds.iaea.or.at/endf/`).

In higher energetic interactions, i.e. for neutron energies above 400 MeV, the production of pions has to be considered. Figure 2.3 shows the cross sections of pion production

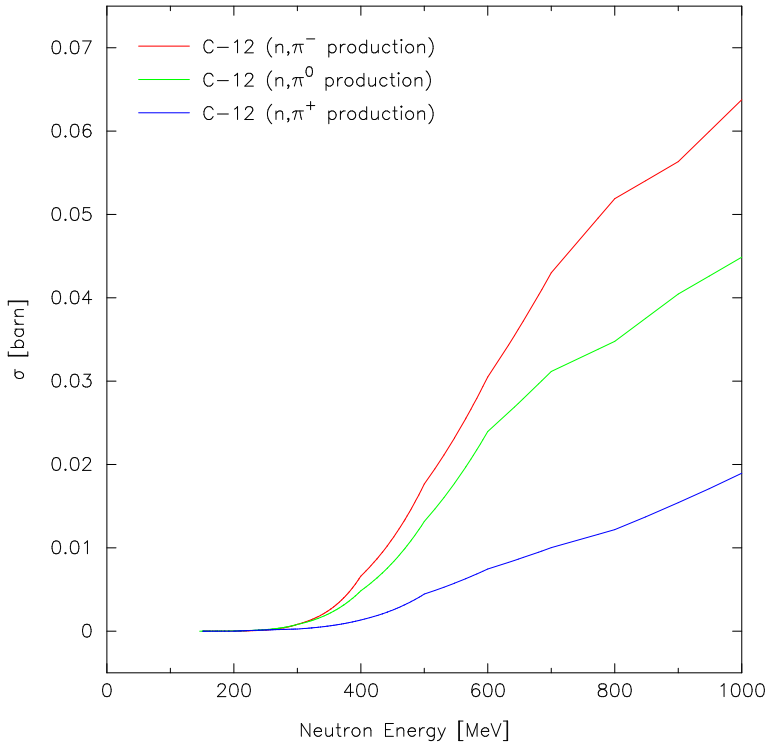


Figure 2.3: Interaction cross sections of the neutron-pion production processes in carbon (from the official ENDF database available at `http://www-nds.iaea.or.at/endf/`).

by neutrons in carbon. The charged pions deposit part or all of their kinetic energy by ionization losses directly in the scintillator. The neutral ones decay within the scintillators according to

$$\pi^0 \rightarrow 2\gamma \rightarrow 2e^+e^-. \tag{2.3}$$

Because of their small mass and the resulting small ionization losses, the electrons and positrons from π^0-decay deposit only a small part of their kinetic energy in the scintillator.

In elastic as well as in inelastic reactions, the recoil protons will receive just a fraction of the kinetic energy of the incident neutron. In a first approximation, the energy exchange is proportional to the neutron energy. On their way through the scintillator the recoil protons deposit energy by excitation and ionization of atoms. These atoms will de-excite under emission of photons, so-called scintillation flashes. The entire produced light is proportional to the energy loss of the protons within the scintillator. In each scintillator unit, the emitted light is amplified by a photomultiplier (2, Figure 2.1). Since the interaction mean free path of protons is much smaller than the depth of the scintillator, the final output signal of the photomultiplier is nearly proportional to the kinetic energy of the recoil protons. This output signal is discriminated at four levels according to the energies $> 40\,\mathrm{MeV}$, $> 80\,\mathrm{MeV}$, $> 120\,\mathrm{MeV}$, and $> 160\,\mathrm{MeV}$. The ability to obtain information on the kinetic energy of the incident neutrons through such a procedure is absolutely necessary for the detailed analysis of solar neutron events (Lockwood et al., 1997; Tsuchiya, 2001).

A scintillation signal can also be generated by incident charged cosmic ray particles such as secondary protons, muons, or electrons. In order to veto charged particles, the

scintillators are surrounded by proportional counters (3, Figure 2.1), 20 at the top, 16 on each side. Neutrons are identified by anticoincidence of the scintillator and the veto counter pulses. The logic of the signal processing is shown in Figure 2.11 and described in Section 2.5.1 below.

Below the scintillators the detector includes two horizontal layers consisting each of 20 proportional counters (4, Figure 2.1). By measuring coincidences between the signal of a counter tube in the upper and in the lower level, it is possible to distinguish between five incident neutron directions (40°N, 25°N, 0°, 25°S and 40°S). The detection principle is illustrated in Figure 2.4. In order to register only the impulses caused by recoil protons,

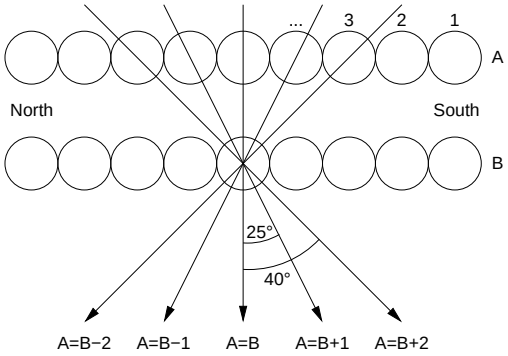


Figure 2.4: Principle of determination of the direction of incidence (cross section of the bottom proportional counters in north-south direction).

only those which coincide with the scintillator signal and anticoincide with the veto counter signal are recorded. The logic is schematically shown in Figure 2.14 below.

The detection of the direction of incidence of the neutrons makes it possible to decide if an increase in the counting rate is due to neutrons from the Sun. In addition, this facility permits to reduce the background caused by secondary neutrons of galactic cosmic rays.

The different mean counting rates of the described experimental setup are summarized in Table 2.1.

	without anti [cts/10 s]	with anti [cts/10 s]
ch1: $>$ 40 MeV	21686	8006
ch2: $>$ 80 MeV	14083	4294
ch3: $>$ 120 MeV	6156	1857
ch4: $>$ 160 MeV	3123	962
Veto counters	61716	
A = B − 2		278
A = B − 1		530
A = B		779
A = B + 1		531
A = B + 2		275

Table 2.1: Mean counting rates of the scintillators and proportional counters in the time period from June 1 to 30, 2002.

2.2 Scintillator Counters

2.2.1 Scintillators

Each of the four 1 m × 1 m × 0.4 m plastic scintillator units consist of a 2 × 2 × 4-matrix of 16 smaller scintillator blocks. The blocks have the size of 0.5 m × 0.5 m × 0.1 m and a weight of about 20 kg.

Polystyrene C_8H_8 is used as scintillating material. A further technical specification of the material is not available to us at the moment. We assume a mean density for plastic scintillators $\rho = 1.032\,\mathrm{g\,cm^{-3}}$. However, all plastic scintillators provide a fast output light signal and have a high light yield (Leo, 1987).

2.2.2 Photomultipliers

Four photomultipliers, one per scintillator unit, are used to amplify the scintillating flashes. To protect the scintillators and photomultipliers from light pollution, they are surrounded by a light-tight cover. The space between the scintillator blocks and the photomultipliers is filled with air.

Type	:	Hamamatsu R1512
Spectral Response Range	:	300 ∼ 650 nm
Dynode Structure	:	Venetian blind
Number of Stages	:	10
Maximum Ratings		
Anode to Cathode Voltage (Vdc)	:	2 kV
Average Anode Current	:	0.1 mA
Cathode Sensitivity		
Quantum Efficency (typ.)	:	26%
Luminous (typ.)	:	90 μA/lm
Blue (min.)	:	8.0 μA/lm-b
Blue (typ.)	:	11.0 μA/lm-b
Anode to Cathode Voltage (Vdc)	:	1.5 kV
Anode Sensitivity		
Luminous (typ.)	:	100 A/lm
Blue (typ.)	:	12 A/lm-b
Anode Dark Current		
Typical	:	20 nA
Maximum	:	100 nA
Typical Current Amplification	:	1.1×10^6
Typical Time Response		
Rise Time	:	15 ns
Electron Transit Time	:	82 ns
Typical Pulse Hight Resolution	:	8.3% NaI (3"⌀×3")
Stability		
Long Term	:	0.2%
Short Term	:	1.0%
Pulse Linearity		
2% Deviation	:	40 mA
5% Deviation	:	80 mA

Table 2.2: Specifications of the photomultipliers (from Hamamatsu data sheet).

The technical specifications of the photomultiplier tubes are given in Table 2.2.

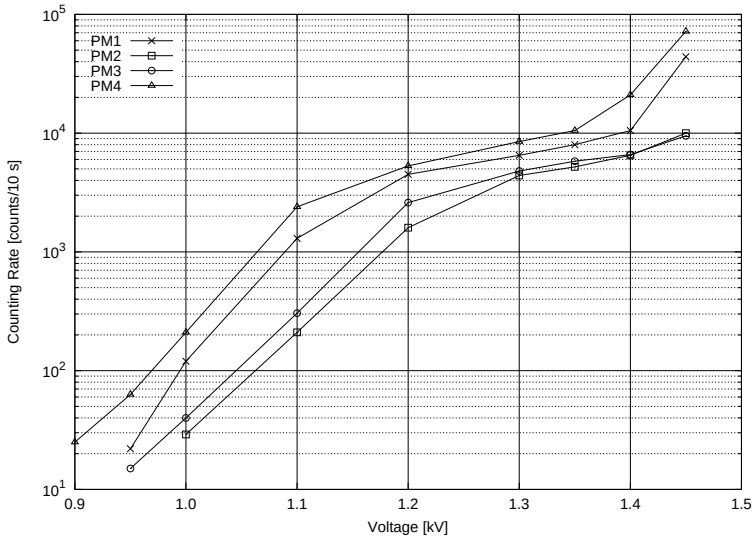


Figure 2.5: Counting rates of the photomultipliers in dependence of operation voltage.

The operation voltage of the photomultipliers is tuned to receive the same counting rate of all four scintillator units while cosmic ray intensity is undisturbed. Table 2.3 shows the current voltage settings. These values were obtained from Figure 2.5 showing the correlation between operation voltage and counting rate.

Section	Operation voltage [kV]
PM1	−1.3
PM2	−1.4
PM3	−1.4
PM4	−1.26

Table 2.3: Operation voltage (cathode to anode) of the photomultipliers.

2.3 Proportional Counters

Around the scintillators there are 124 proportional counter tubes operating either as veto counters to identify charged particles or as directional counters to identify the incident direction of particles.

Location	section (counters per section)
top from front (west) to back (east)	E1 (5), E2 (5), B1 (5), B2 (5)
back (east) from top to bottom	C2 (3), C1 (3), D2 (5), D1 (5)
front (west) from top to bottom	A2 (3), A1 (3), F2 (5), F1 (5)
left (north) from to to bottom	G (6), H (10)
right (south) from top to bottom	I (6), J (10)

Table 2.4: Denotation of veto counter sections.

The counter tubes acting as directional counters are arranged in four sections as defined in Table 2.5. The distance between the layers (from center to center of the tubes) is 20 cm.

Location	counter (counters per section)
upper layer A from left (north) to right (south)	L10 ... L1 (10), K10 ... K1 (10)
lower layer B from left (north) to right (south)	N10 ... N1 (10), M10 ... M1 (10)

Table 2.5: Denotation of the directional counters and counter sections.

According to Figure 2.4 this arrangement defines the five directions of incidence as listed in Table 2.6.

Denotation	Direction
$A = B + 2$	40°N
$A = B + 1$	25°N
$A = B$	0°
$A = B - 1$	25°S
$A = B - 2$	40°S

Table 2.6: Denotation of the 5 incident directions.

The technical specifications of the used tubes are listed in Table 2.7.

Diameter		
outside	:	10.1 cm
inside	:	9.75 cm
Length		
Sections A, B, C, D, E, F	:	250 cm
Sections G, H, I, J, K, L, M, N	:	200 cm
Tube material	:	Stainless steel
Counter gas	:	P10 (90% Ar, 10%CH_4)
Gas pressure		
Sections A, B, C, D, G, H, I, J, K, L, M, N	:	0.8 atm
Sections E, F	:	1.0 atm
Diameter of anode wire		
Sections A, B, C, D, G, H, I, J, K, L, M, N	:	50 μm
Sections E, F	:	100 μm

Table 2.7: Specifications of the proportional counters.

The electronic circuits of the high-voltage device and of the preamplifier/discriminator units are shown in Figures 2.6 and 2.7.

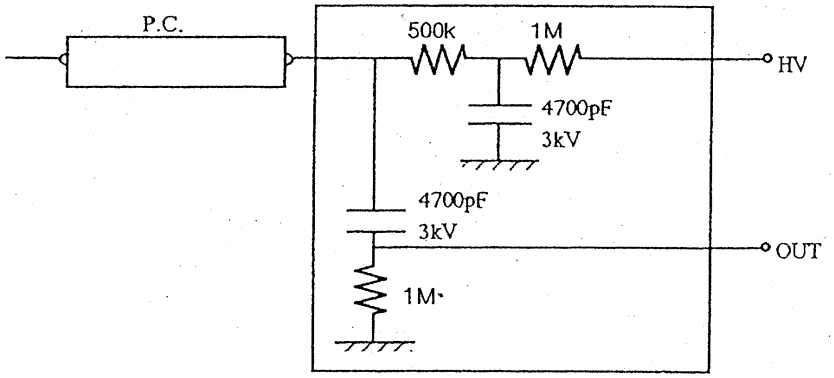


Figure 2.6: The high-voltage device of the proportional counters.

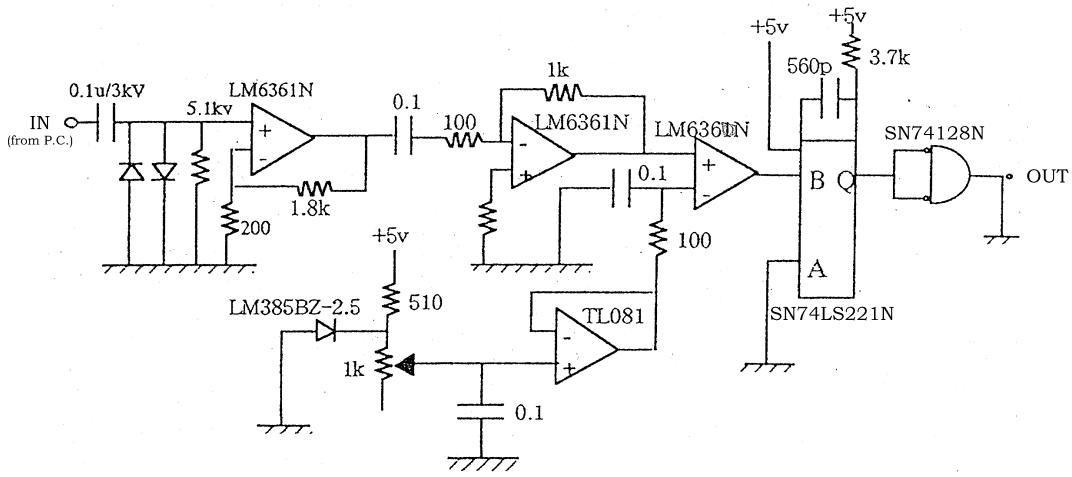


Figure 2.7: The preamplifier/discriminator circuit.

In order to register charged particles, the proportional counters are operated in the Geiger-Müller mode. Figure 2.8 shows the counter characteristics of a part of the sections. The currently applied operation voltages for all sections are listed in Table 2.8.

Section	A	B	C	D	E	F	G
Voltage [kV]	2.4	2.44	2.4	2.38	2.6	2.54	2.4
Section	H	I	J	K	L	M	N
Voltage [kV]	2.4	2.4	2.4	2.4	2.4	2.4	2.4

Table 2.8: Operation voltages of the counter sections.

Table 2.9 shows the operation voltage and current of the 56 preamplifier/discriminator units (16 used for veto, 40 for directional counters). These values were determined from measurements of the impulse height distribution and are characterized by the parameters given in Table 2.10.

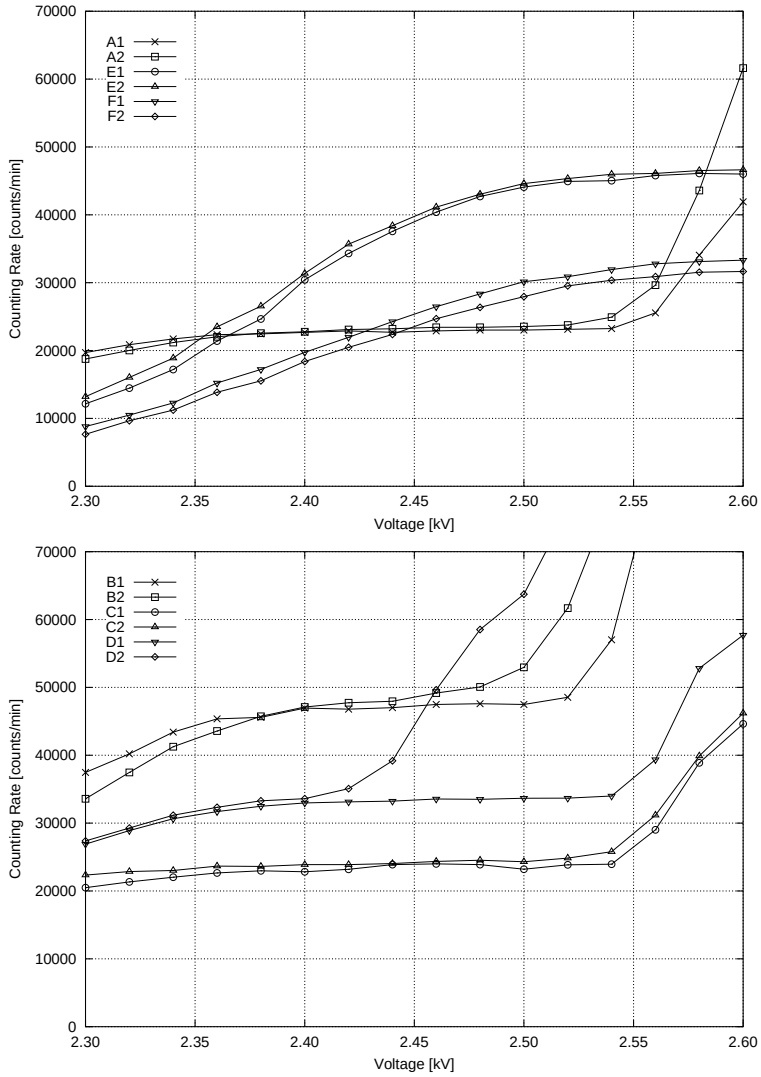


Figure 2.8: Counter characteristics of sections A, B, C, D, E, and F.

Operation voltage [V]	Current per unit [mA]	Current total [A]
+5	80	4
−5	20	1

Table 2.9: Operation voltage, current per preamplifier/discriminator unit and total current.

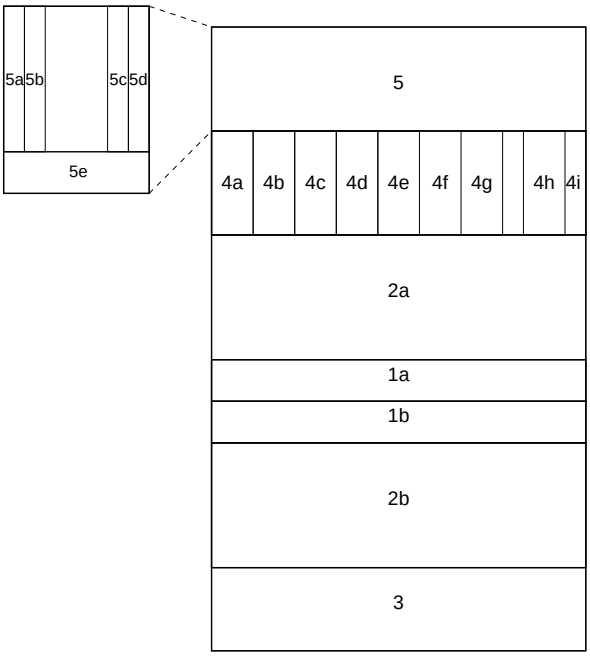
Section	Slope [%/mV]	Discriminator-voltage [mV]
A1	0.003	500
A2	0.001	500
F1	0.05	400
F2	0.01	400

Table 2.10: Spectrum of impulse height: Slope of the plateau at the discrimination level.

2.4 Power Supplies

2.4.1 High Voltage

Two devices *Spellman SL 150* are used as high-voltage power supplies for the proportional counters and the photomultipliers. They are arranged vertically within a rack as illustrated in Figure 2.9, where the upper one (1a) supplies the counter tubes and the lower one (1b) the photomultipliers. The specifications of the devices are shown in Table 2.11 and the current settings in Table 2.12.



1 High-Voltage Power Supplies
  a) for Proportional Counters
  b) for Photomultipliers
2 High-Voltage Dividers
  a) for Proportional Counters
  b) for Photomultipliers
3 Low-Voltage Power Supply
4 NIM Plug-in Unit
  a) Amplifier
  b) First Discriminator
  c) Second Discriminator
  d) OR-Gate Module
  e) Original Circuit
  f) TTL-to-NIM-Converter
  g) Thermometer/Barometer
  h) Direction Module
  i) NIM Power Supply
5 CAMAC-Crate
  a) Left Scaler Unit
  b) Right Scaler Unit
  c) Dataway Indicator
  d) CRATE-Controller
  e) CRATE-Power Supply

Figure 2.9: Schematic front view of the rack.

Input	:	100 V ± 10%, 50/60 Hz
Output	:	max. 3 kV, 50 mA, 150 W
Ripple	:	0.1% p–p of maximum output
Temperature Coefficient	:	100 ppm/°C voltage or current regulated
Ambient Temperature	:	0°C to 50°C
Stability	:	100 ppm/hour after 1/2 hour warm-up

Table 2.11: Specifications of the high-voltage supplies *Spellman SL 150* (from Spellman data sheet).

Supply	Voltage [kV]	Current [mA]
Counter tubes	+2.60	4.8
Photomultipliers	−1.40	3.8

Table 2.12: Settings of the high-voltage supplies *Spellman SL 150*.

2.4.2 High-Voltage Dividers

To divide the high voltage for the proportional counters and the photomultipliers, two devices *10CH High Voltage Divider Model S-NS201* manufactured by *Techno Land* are being used. The current settings of the voltage divider of the counter tubes (2a, Figure 2.9) are shown in Table 2.13 and the ones of the photomultipliers (2b) in Table 2.14. These tables include also the present relationship between channel and detector section.

Channel	Setting [V]	Operation voltage [kV]	Section
1	200	2.4	A, C, G, H, I, J
2	0	2.6	E
3	140	2.46	B
4	defect	2.6	–
5	220	2.38	D
6	100	2.5	F
7	200	2.4	K
8	200	2.4	L
9	200	2.4	M
10	200	2.4	N

Table 2.13: Settings of the high-voltage divider *S-NS201* of the counter tubes.

Channel	Setting [V]	Operation voltage [kV]	Section
1	100	−1.3	PM1
2	0	−1.4	PM2
3	0	−1.4	PM3
4	140	−1.26	PM4

Table 2.14: Settings of the high-voltage divider *S-SN201* of the photomultipliers.

2.4.3 Low Voltage

The preamplifier/discriminator units of the proportional counters are powered by a *Power Supply PS-5246A* (3, Figure 2.1) containing different *ELCO G Series Power Supply* modules. The specifications of these modules are listed in Table 2.15 and the present settings in Table 2.16.

ELCO Module	GT3.5 12V 4.5A	GT3 5V 5A	G1 −6V 1A
Current	$0 \sim 4\,\mathrm{A}$	$0 \sim 4\,\mathrm{A}$	$0 \sim 0.5\,\mathrm{A}$
Voltage	$10 \sim 14\,\mathrm{V}$	5 V	$-5 \sim -7\,\mathrm{V}$
Overflow	4.6 A	4.4 A	0.6 A
Stability on Load	0.1%	±1.5%	±0.2%
Stability (AC $100\,\mathrm{V} \pm 10\%$)	0.1%	±1.5%	±0.2%
Ripple p–p	5 mV	50 mV	5 mV
Overload Voltage		7 V	

Table 2.15: Specifications of power supply *PS-5246A*.

ELCO Module	Operation voltage [V]	Current [A]
GT3.5 12V 4.5A	+12.09	0.0
GT3 5V 5A	+6.12	4.3
G1 −6V 1A	−4.92	not in use

Table 2.16: Settings of power supply *PS-5246A*.

2.5 Signal processing

2.5.1 NIM

For the signal processing, *NIM* (Nuclear Instrumentation Methods) modules manufactured by *Techno Land* are being used. They are horizontally arranged within the NIM plug-in unit as shown in Figures 2.9 and 2.10.

The output signals of the four photomultipliers are amplified by the module *N-TM230* (4a, Figure 2.9) and afterwards discriminated at four levels (energy channels ch1 to ch4) by two different modules. Module *N-TM104* (4b) discriminates ch1 (> 40 MeV) and ch2 (> 80 MeV) while module *N-TM405* (4c) is discriminating ch3 (> 120 MeV) and ch4 (> 160 MeV). The settings of the discrimination levels are summarized in Table 2.17. The four

Channel	Discrimination voltage [mV]
ch1	−75
ch2	−150
ch3	−225
ch4	−300

Table 2.17: Settings of the discrimination levels.

signals of each energy channel are treated by the OR-gate module *N-SE211* (4d), and the logic output of each gate is connected to the left CAMAC scaler unit (5a, channels "PM without anti") and to the module *Original Circuit* (4e). The output signals of the veto sections are also connected to this last module to determine the anticoincidence between the energy channels and the veto signals. The TTL output signal of *Original Circuit* is

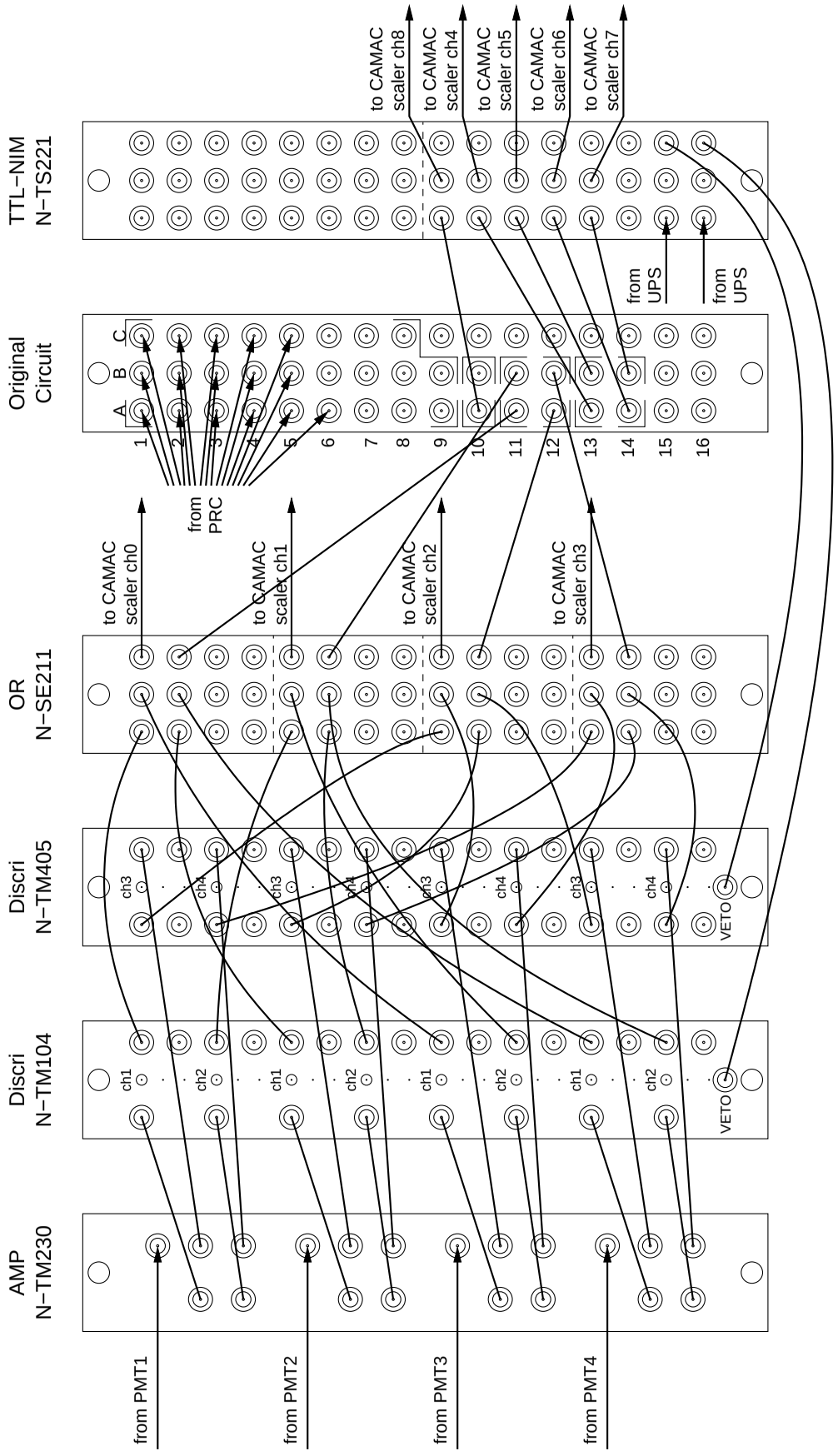


Figure 2.10: NIM modules and cabling for signal processing.

converted back into NIM standard by the *TTL-to-NIM-Converter N-TS221* (4f) and is finally recorded by the left CAMAC scaler unit (5a, channels "PM with anti"). The logic circuit of the neutron detector is shown schematically in Figure 2.11, the electronic circuit of *Original Circuit* in Figure 2.12.

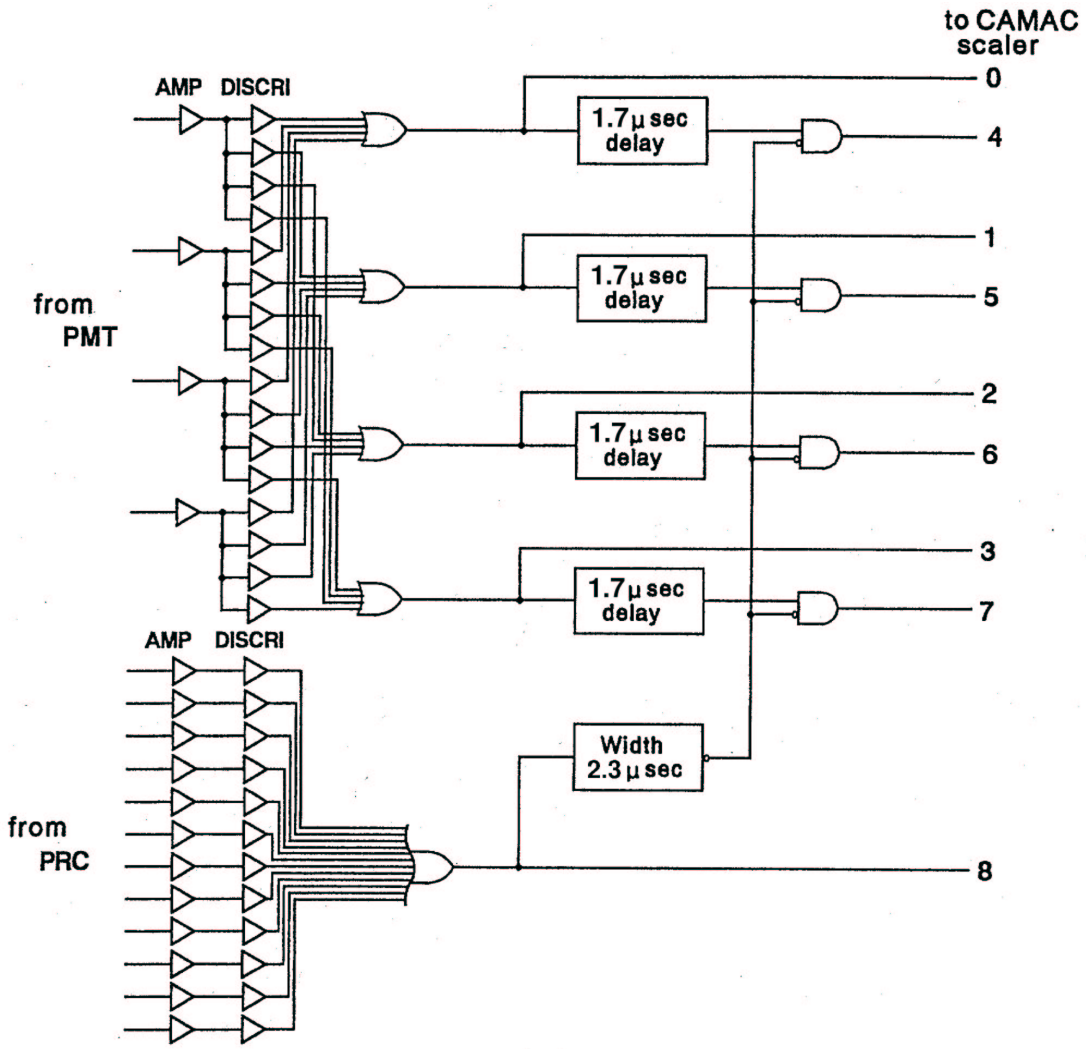


Figure 2.11: Logic circuit of the neutron detector (without detection of direction).

An additional port of the TTL-to-NIM-Converter converts a logic output of the uninterruptible power supply (UPS). The corresponding NIM output is connected to the ports *VETO* of the discriminator modules. The logic signal of the UPS will thus set all counting rates to zero if the line voltage breaks down.

The electronic circuit for the detection of the direction of incidence is also placed within a module in the NIM plug-in unit (4h). Its front view with port allocation is shown in Figure 2.13.

The logic can discriminate between five different directions of incidence by comparing simultaneous pulses of a counter tube in the upper and in the lower layer of sections K, L, M, and N, respectively. The module vetoes charged primary particles by measuring coincidence with *Original Circuit* output A13 (PM ch1: > 40 MeV with anti), i.e. antico-

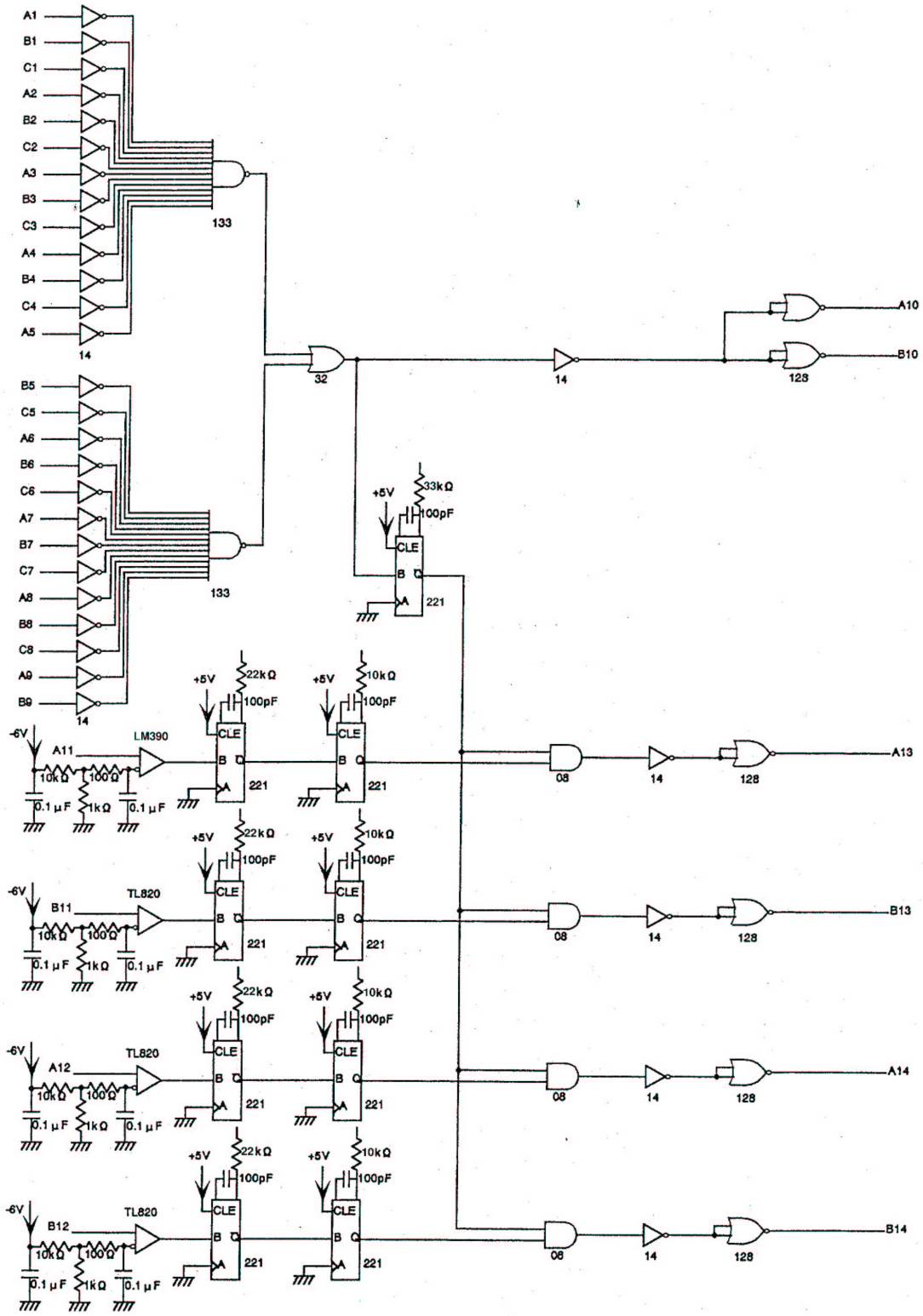


Figure 2.12: Electronic circuit of the module *Original Circuit*.

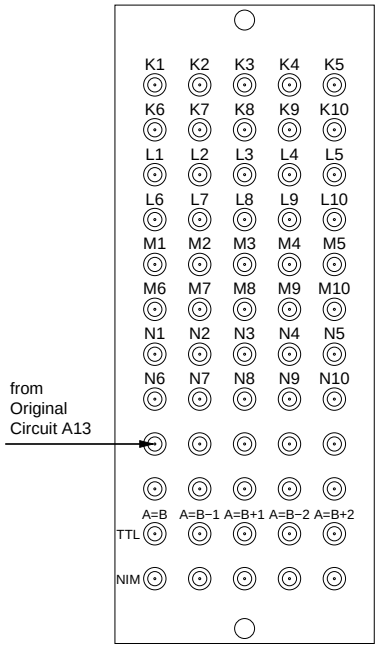


Figure 2.13: Cabling and port allocation of the direction module.

incidence with veto counters and coincidence with scintillators.

As an example of the logics, Figure 2.14 shows schematically the signal processing for the direction $A = B - 1$. The logic for the other directions is defined similarly by pairwise linking of the input signals according to $A = B - 2$, $A = B + 2$, $A = B + 1$, and $A = B$, respectively.

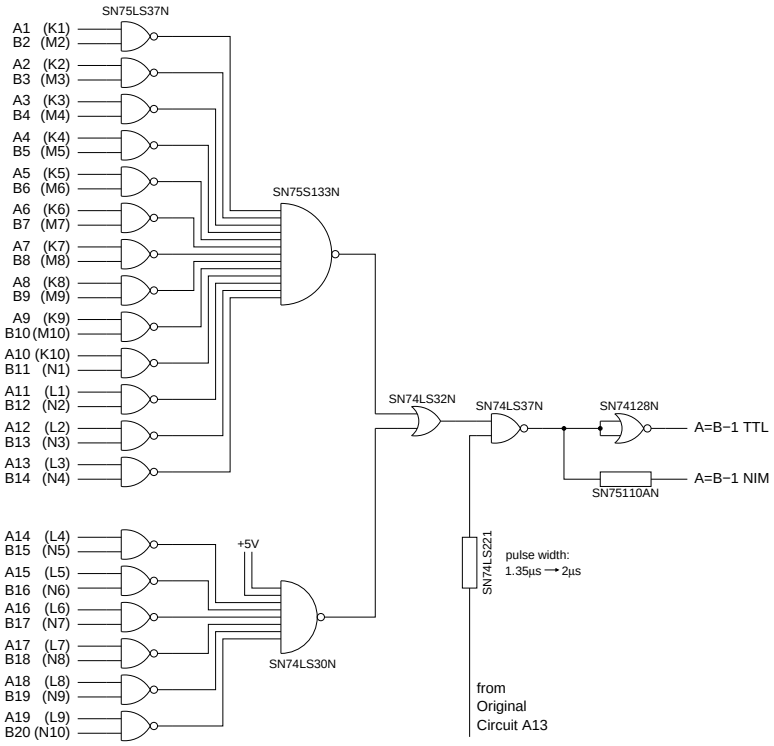


Figure 2.14: Logic circuit of the direction $A = B - 1$.

2.5.2 Temperature and Pressure Measurement

The thermometer/barometer is also installed within the NIM plug-in unit (4g, Figure 2.9). The two output signals *temperature frequency* f_T and *pressure frequency* f_p are connected to the ports 14 and 15, respectively, of the left CAMAC scaler unit.

The temperature T and atmospheric pressure p inside the experiment container must be calculated later from the two counting rates f_T [s^{-1}] and f_p [s^{-1}]. According to the manufacturer the temperature is given by

$$X\,[\mu s] = 10^6/f_T, \tag{2.4}$$

$$U\,[\mu s] = X - U_0, \tag{2.5}$$

$$T\,[^\circ C] = Y_1 U + Y_2 U^2 \tag{2.6}$$

and the pressure by

$$V\,[\mu s] = 10^6/f_p, \tag{2.7}$$

$$C\,[\text{psia}] = C_1 + C_2 U + C_3 U^2, \tag{2.8}$$

$$V_0\,[\mu s] = V_1 + V_2 U + T_3 U^2 + T_4 U^3, \tag{2.9}$$

$$p\,[\text{psia}] = C_1 - T_0^2/T_1^2 - D(1 - T_0^2/T^2), \tag{2.10}$$

or, in other units

$$p\,[\text{hPa}] = 68.95 \times p\,[\text{psia}], \tag{2.11}$$

$$p\,[\text{mmHg}] = 51.905 \times p\,[\text{psia}]. \tag{2.12}$$

Table 2.18 contains the coefficients.

U_0	[μs]	:	5.889628
Y_1	[°C/μs]	:	−39030.678000
Y_2	[°C/μs^2]	:	−9613.669000
C_1	[psia]	:	96.829780
C_2	[psia/μs]	:	3.787943
C_3	[psia/μs^2]	:	−106.228900
D	[psia]	:	0.036680
T_1	[μs]	:	27.844050
T_2	[μs/μs]	:	0.720212
T_3	[μs/μs^2]	:	18.315750
T_4	[μs/μs^3]	:	32.646550

Table 2.18: Coefficients for temperature and pressure determination.

2.5.3 CAMAC

Two *CAMAC-Scaler* units *C-TS112* (5a,b, Figure 2.9) by *Techno Land* are located in the CAMAC-Crate. Table 2.19 shows the port allocation of the input signals of the scalers.

The output signals are connected to the *Dataway Indicator C-TM208* (5c) and further to the *CRATE-Controller CCP-F* (5d). There the data can be read out via a CAMAC-standard linking cable and data-taking card by a data-taking computer.

Port	left (5a)	right (5b)
0	PM ch1 without anti	PR section A1
1	PM ch2 without anti	PR section A2
2	PM ch3 without anti	PR section B1
3	PM ch4 without anti	PR section B2
4	PM ch1 with anti	PR section C1
5	PM ch2 with anti	PR section C2
6	PM ch3 with anti	PR section D1
7	PM ch4 with anti	PR section D2
8	total veto PR counters	PR section E1
9	PR direction A = B	PR section E2
10	PR direction A = B − 1	PR section F1
11	PR direction A = B + 1	PR section F2
12	PR direction A = B − 2	PR section G
13	PR direction A = B + 2	PR section H
14	temperature frequency [1/10 s]	PR section I
15	pressure frequency [1/10 s]	PR section J

Table 2.19: Port allocation of the CAMAC-Scalers C-TS 112.

2.6 Data-Taking System

The on-line data-taking system is schematically shown in Figure 2.15. Two different computers are in use for data-taking, *sonete2.unibe.ch* (IP *130.92.230.51*) and *sonete1.unibe.ch* (IP *130.92.230.50*). Furthermore a GPS receiver and an EXAbyte storage device are connected to them.

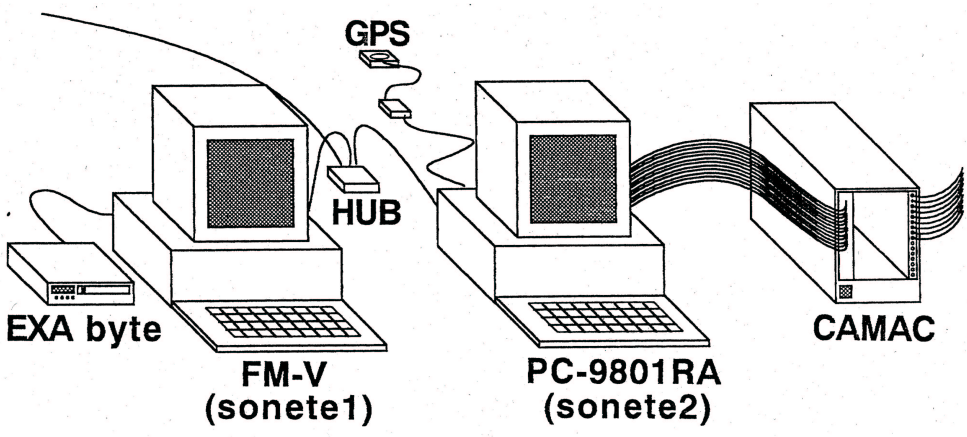


Figure 2.15: Arrangement of the data-taking devices (after Flückiger et al., 1998).

The computer *sonete2.unibe.ch*, a *PC-9801RA* manufactured by *NEC* is operated by the *MS-DOS 5.0* operating system. This computer executes mainly the data-taking software. The different counting rates are read out every 10 seconds from the CRATE-Controller and stored on the hard disc of *sonete1.unibe.ch*. Additionally all the data can be visualized on the display.

With each data set two different times are stored. The first one corresponds to the system clock of *sonete2.unibe.ch* which is regularly synchronized with *sonete1.unibe.ch*. The other time is taken directly from the GPS receiver, which inside the container, however, does not receive a signal from the GPS satellites.

The other computer *sonete1.unibe.ch*, a *FM-V DESKPOWER* by *FUJITSU*, is operated by *Solaris 2.5.1 for Intel Platform* disc operating system. The system clock is synchronized with the NTP timeserver *sns2-tss2.unige.ch* once a day. The current time is also saved every second in a file which is used by *sonete2.unibe.ch* to synchronize its system clock. All the saved data on *sonete1.unibe.ch* can be downloaded from Bern or Nagoya via the installed FTP server. For safety reasons the data file of the past day is written once a day on the EXAbyte tape.

2.7 Barometer Coefficients

The incident cosmic ray intensity varies with atmospheric pressure. In order to evaluate pure variations of the cosmic ray intensity, the counting rate N has to be corrected for pressure changes. Corrections are made according to

$$N_{\mathrm{corr}} = N\,\mathrm{e}^{\alpha(p-\bar{p})}, \tag{2.13}$$

where α is the barometer coefficient and $\bar{p} = 514\,\mathrm{mmHg}$ the mean pressure at Gornergrat.

The different barometer coefficients α of the energy channels with and without anti of the veto and the directional counters were determined experimentally. They were calculated using the equations of Section 3.2 in Bütikofer (1988). The time intervals to determine the barometer coefficients were chosen according to the following criteria:

- The fluctuations of the primary cosmic ray flux are only small within the time interval.
- $|\Delta p| > 3.5\,\mathrm{mmHg}$.
- $|\Delta p/\Delta t| > 0.15\,\mathrm{mmHg/h}$.
- The duration of the interval is chosen for 24 hours or a multiple of it, so that the influence of the diurnal variation of cosmic ray intensity is minimal.

The obtained values are listed in Table 2.20.

	without anti α [%/mmHg]	with anti α [%/mmHg]
ch1: $>$ 40 MeV	0.56	0.70
ch2: $>$ 80 MeV	0.51	0.67
ch3: $>$ 120 MeV	0.59	0.72
ch4: $>$ 160 MeV	0.76	0.86
Veto counters	0.70	
A = B − 2		0.505
A = B − 1		0.485
A = B		0.520
A = B + 1		0.485
A = B + 2		0.505

Table 2.20: Barometer coefficients of SONTEL at Gornergrat.

Chapter 3

Cosmic Rays in the Earth's Atmosphere

Most of the cosmic ray particles that reach the Earth's surface are of secondary generation. Primary cosmic ray particles, mostly protons, interact with nuclei of atmospheric constituents leading to a cosmic ray cascade as illustrated in Figure 3.1. In collisions of secondaries with atmospheric target nuclei, nuclear excitation and evaporation of target nuclei may occur. Unstable particles will either decay or cause further interactions. Electrons and photons contribute to the cascade via bremsstrahlung and pair production, respectively.

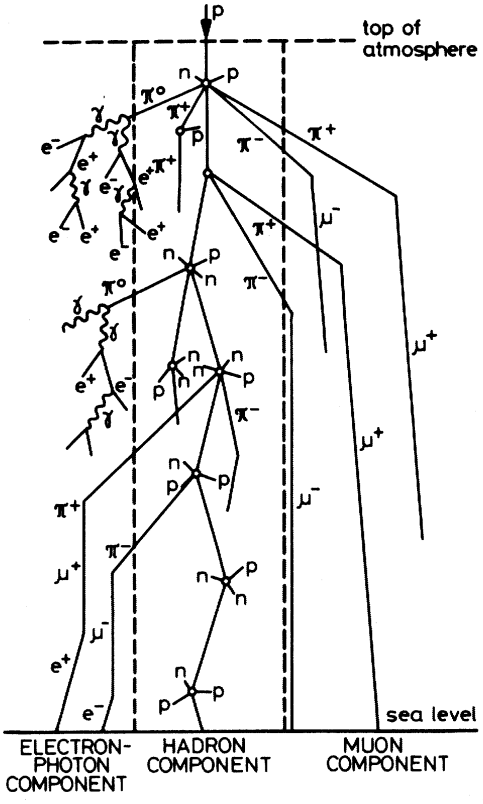


Figure 3.1: Cosmic ray cascade: Schematic representation of particle production in the atmosphere triggered by a primary cosmic ray proton. For simplicity, neutrinos resulting from various decays are not shown. (From Allkofer and Grieder, 1984)

In order to simulate the SONTEL detector at Gornergrat altitude level, the secondary cosmic ray spectra due to the galactic component of cosmic radiation must be known. In particular quantitative information is needed about the most abundant types of particles above 40 MeV and at the atmospheric depth of $700\,\mathrm{g\,cm^{-2}}$. Even though these particle spectra could also be simulated by the Monte Carlo technique described in this thesis, the corresponding spectra are taken from literature. Some of them are simulation results, the others are based on measurements.

Where necessary, the spectra had to be adapted to the atmospheric depth of Gornergrat. The altitude correction of the vertical intensity $I_i(0°, X_1)$ of the nucleonic component can be made according to

$$I_i(0°, X_2) = I_i(0°, X_1) \cdot \exp\left(-\frac{X_2 - X_1}{\Lambda_i}\right), \tag{3.1}$$

where X_1 is the atmospheric depth corresponding to the original spectrum, X_2 the atmospheric depth of Gornergrat, and Λ_i the attenuation length of the particle species i. The intensity $I_i(\theta, X_2)$ for non-vertical directions given by the zenith angle θ is usually expressed by

$$I_i(\theta, X_2) = I_i(0°, X_2) \cdot \cos^{n_i}(\theta), \tag{3.2}$$

where the exponent $n_i = n_i(T, X)$ is a characteristic of the individual particle type i.

In the following paragraphs we describe for $X_2 = 700\,\mathrm{g\,cm^{-2}}$ the differential energy spectra of the four most abundant particle types, neutrons, protons, muons, electrons and of the γ-radiation. The *differential vertical energy spectrum* $j_i(T)$ of any kind of component i is defined as

$$j_i(T) = \frac{\mathrm{d}N_i(T)}{\mathrm{d}A\,\mathrm{d}t\,\mathrm{d}\Omega\,\mathrm{d}T}\ \ [\mathrm{cm^{-2}s^{-1}sr^{-1}MeV^{-1}}], \tag{3.3}$$

where $\mathrm{d}N_i(T)$ is the number of particles or photons, per unit area $\mathrm{d}A$, per time unit $\mathrm{d}t$, per unit solid angle $\mathrm{d}\Omega$, and per interval $\mathrm{d}T$ of kinetic energy.

3.1 Neutrons

As it is not easy to measure the energy spectra of neutrons, hardly any reliable data sets are available. Flückiger (1976) simulated the atmospheric nucleon cascade and determined the neutron spectrum by a Monte Carlo simulation. Results are available for an atmospheric depth of $X_1 = 650\,\mathrm{g\,cm^{-2}}$ and for mean solar activity. Figure 3.2 shows the vertical differential neutron spectrum corrected to Gornergrat altitude according to (3.1) with an attenuation length $\Lambda_\mathrm{n} = 150\,\mathrm{g\,cm^{-2}}$.

The exponent of the cosine law (3.2) to approximate the zenith angle dependent flux is given by Grieder (2001, p. 336) as $n_\mathrm{n} = 3.5$.

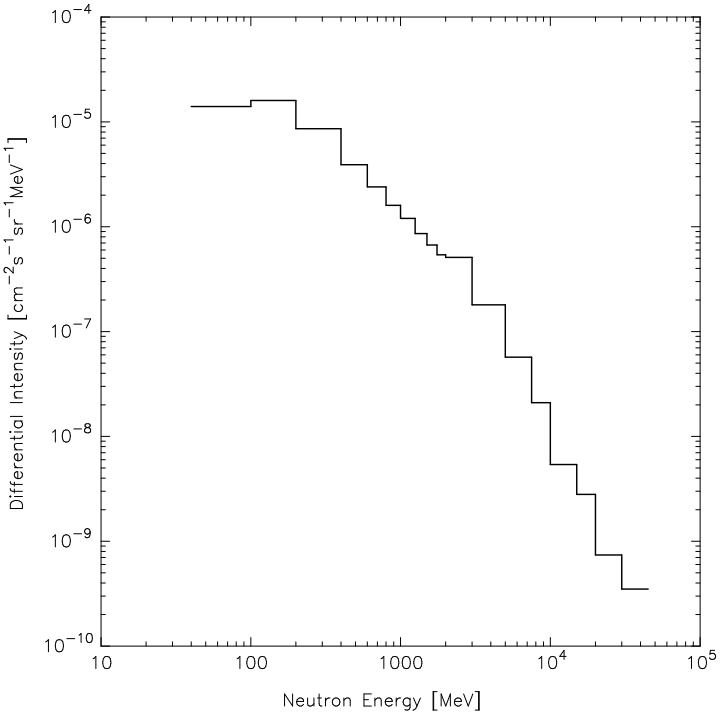


Figure 3.2: Differential vertical energy spectrum of neutrons for mean solar activity and at the atmospheric depth $X_2 = 700\,\mathrm{g\,cm^{-2}}$.

3.2 Protons

Flückiger (1976) also determined the proton spectrum at the atmospheric depth of $X_1 = 650\,\mathrm{g\,cm^{-2}}$ and for mean solar activity. This spectrum was altitude corrected with an attenuation length $\Lambda_\mathrm{p} = 140\,\mathrm{g\,cm^{-2}}$. Figure 3.3 shows the resulting differential vertical

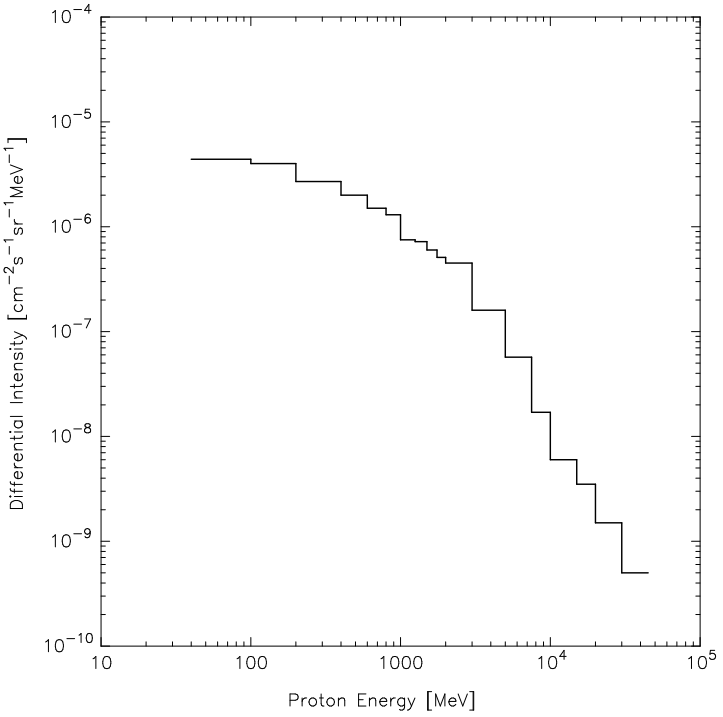


Figure 3.3: Differential vertical energy spectrum of protons for mean solar activity and at the atmospheric depth $X_2 = 700\,\mathrm{g\,cm^{-2}}$.

energy spectrum of the protons.

The zenith angle dependence is approximated by (3.2) with an exponent $n_\mathrm{p} = 5.7$ (Grieder, 2001, Fig. 3.7).

3.3 Muons

Due to the specific production mechanism of the muons in the atmosphere, equation (3.1) cannot be used to describe the dependence of the muon intensity on atmospheric depth. We interpolated linearly two momentum spectra from Grieder (2001, Fig. 2.137) at two different atmospheric depths to obtain the energy spectrum at $X_2 = 700\,\mathrm{g\,cm^{-2}}$. This spectrum is shown in Figure 3.4. Because of the production mechanism of the muons in

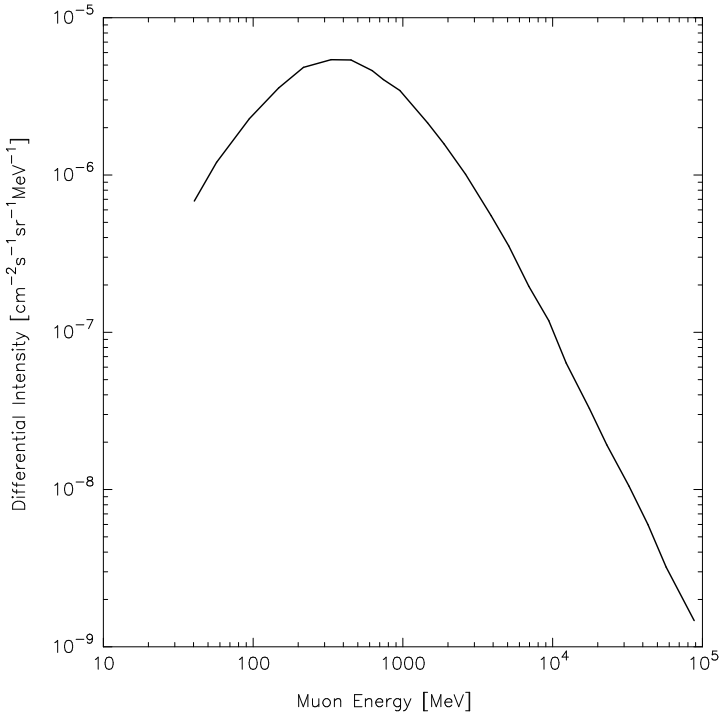


Figure 3.4: Differential vertical energy spectrum of muons at the atmospheric depth $X_2 = 700\,\mathrm{g\,cm^{-2}}$.

the atmosphere the intensity is not isotropic with respect to azimuth. However, for the following calculations the dependence on azimuth angle can be neglected.

For the description of the zenith angle dependent intensity, we use the cosine exponent $n_\mu = 1.85$ according to Allkofer and Grieder (1984).

3.4 Other Components

Neutrons, protons, and muons are the main contributors to the counting rates of SONTEL. However, the contributions by gamma rays and electrons are not negligible. Therefore, for all the simulations conducted within this thesis these two components are taken into account.

The vertical gamma energy spectrum is taken from Grieder (2001, Fig. 3.20), as well as the exponent $n_\gamma = 7.3$ defining the zenith angle dependence according to (3.2). The total

spectrum of the electrons and positrons is also obtained from the same source (Fig. 3.24). According to Grieder (2001, Fig. 2.105) the electron intensity at Gornergrat is about 3 times higher than at sea level. This result is used for the altitude correction. The zenith angle dependence of the electron intensity is described by a cosine exponent $n_e = 3.6$. Both spectra are shown in Figure 3.5.

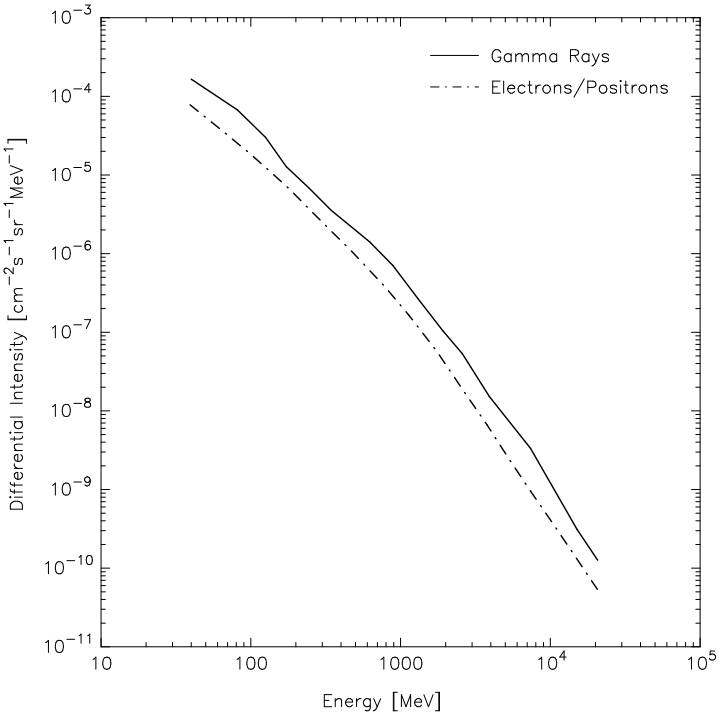


Figure 3.5: Differential vertical energy spectra of the gamma ray and the electron/positron component at the atmospheric depth $X_2 = 700\,\mathrm{g\,cm^{-2}}$.

As illustrated in Figure 3.1 there is also a pion component in the atmospheric cosmic ray cascade. Below $\sim 10\,\mathrm{GeV}$ the ratio between charged pions and protons is of the order of ten percent. Therefore, the influence of the pions in the present work is neglected.

3.5 Integral Intensities

The *integral vertical intensity* $I_i(0°, X_2)$ for the particle type i, which is used as an input parameter in our simulation, is defined as

$$I_i(0°, X_2) = \int_{T_{\min}}^{T_{\max}} j_i(T)\,\mathrm{d}T \;\; [\mathrm{cm^{-2}s^{-1}sr^{-1}}], \tag{3.4}$$

where $j_i(T)$ are the differential vertical spectra discussed above.

Integration has been made numerically. The lower limit of the energy range was set to $T_{\min} = 40\,\mathrm{MeV}$, since lower energies cannot be measured by the photomultipliers. The upper limit was arbitrarily set to $T_{\max} = 20\,\mathrm{GeV}$, because secondary particles with higher kinetic energies are too rare to be significant. The results of the integration are given in Table 3.1.

Particle	$I(0°, X_2 = 700\,\mathrm{g\,cm^{-2}})$ $[\mathrm{cm^{-2}s^{-1}sr^{-1}}]$
Neutron	1.484×10^{-2}
Proton	2.706×10^{-3}
Muon	2.500×10^{-2}
Gamma	9.895×10^{-3}
Electron	4.293×10^{-3}

Table 3.1: Integral vertical cosmic ray intensities in the energy range $40\,\mathrm{MeV} \leq T \leq 20\,\mathrm{GeV}$ at the atmospheric depth $X_2 = 700\,\mathrm{g\,cm^{-2}}$.

Chapter 4

Monte Carlo Applications

To be able to interpret the measurements of any particle detector, it is necessary to know in detail the correlation between the incident particle flux and the resulting counting rates.

Tsuchiya et al. (2001) determined experimentally the neutron detection sensitivity of SONTEL in an accelerator beam. Since the original detector was too large to be exposed to an accelerator beam, they used a miniaturized version (Matsubara et al., 1997). In this model detector only the major components, i.e. the scintillators and the proportional counters, could be considered. The influence of the real geometric arrangement of the original detector parts is therefore still unknown.

It is the main goal of this work to obtain the detection properties of the SONTEL detector at Gornergrat theoretically. Because of the complex detector geometry and the large number of different interaction mechanisms, it is quite impossible to derive the detector properties analytically. Instead, the unknown parameters are obtained by a Monte Carlo simulation. The procedure to derive a complete set of detector parameters consists of three parts.

1. A Monte Carlo application computes the response of SONTEL to the secondary galactic cosmic ray background. The various counting rates due to the different components are simulated and the basic instrumentation properties determined.

2. Before the detector response to solar protons and neutrons can be calculated, the secondary particle spectra of a solar event at the atmospheric depth of Gornergrat have to be known. Therefore, another application calculates the corresponding spectra from given solar proton and neutron input spectra at the top of the atmosphere.

3. The secondary spectra obtained in part 2 are used by a third Monte Carlo application in order to simulate the increase of the various counting rates due to a solar event.

The following sections give an introduction to the existing software libraries which can be used to realize the corresponding program applications. Furthermore, the basic principles and the use of the applications developed for this work are explained.

4.1 Software

In order to simulate the function of a complex detector like SONTEL, a Monte Carlo code is needed which must fulfill at least the following different requirements:

- A radiation source must irradiate the detector from discrete directions with a specific type of radiation and a predefined energy distribution.
- The whole detector geometry including the properties of all the used materials must be specified.
- All the relevant physical interaction processes between radiation and matter in the corresponding energy range must be simulated.
- For each part of the detector the energy deposit must be recorded.
- The signal processing must be implemented.

These requirements can only be fulfilled within a reasonable time if one can make use of predefined libraries. In particular, the models of physical interaction processes are theoretically so advanced that they cannot be understood in every detail.

For this purpose, a few Monte Carlo codes including most of the available physical models needed for a numerical detector simulation are available. Each of these software packages provides the implementation of (idealized) detectors in its own way. The following paragraphs give a short overview of some of these simulation tools.

4.1.1 GEANT3

GEANT3 is part of the widespread CERN Program Library (CERNLIB), a large collection of general purpose programs maintained and offered in both source (FORTRAN77) and object code form. Most of these programs were developed at CERN and are therefore oriented toward the needs of a physics research laboratory but are also applicable to a wide range of other problems.

Originally designed for the simulation of high energy physics detectors, GEANT3 has today fewer applications in its original field. The implemented physical models are not able to describe the physical interaction processes of particles at the energies reached by modern particle accelerators. Therefore, a successor was developed, and GEANT3 will no longer be maintained after 2002.

All the software and documentation are available free at the CERN web pages:

- CERN Program Library Homepage:
  `http://wwwinfo.cern.ch/asd/cernlib`
- Latest versions of CERNLIB:
  `http://wwwinfo.cern.ch/asd/cernlib/version.html`
- Documentations of CERNLIB components:
  `http://wwwinfo.cern.ch/asdoc/Welcome.html`

4.1.2 GEANT4

In order to simulate forthcoming high energy physics experiments at the Large Hadron Collider (LHC) at CERN, a successor to GEANT3 was needed. The idea was to develop a Monte Carlo code from scratch, but in the object-oriented programming language C++. Existing models describing the physics needed to be improved or replaced by others to become valid for higher energies. The first version of the newly designed GEANT4 was released in December 1998.

All the required source code, libraries, databases, and utilities are available free on the following web pages:

- GEANT4 C++ source code, databases, and installation instructions:
  `http://cern.ch/geant4`
- Graphical user interfaces GGE, GPE and GAG for Java:
  `http://erpc1.naruto-u.ac.jp/~geant4/`
- Visualization tools DAWN, DAWNCUT, and DAVID:
  `http://geant4.kek.jp/~tanaka/`

Documentation and further information about the Monte Carlo code and the implemented physics models are available on the following web sites:

- Installation Guide: For setting up Geant4 in your computing environment:
  `http://wwwinfo.cern.ch/asd/geant4/G4UsersDocuments/UsersGuides/InstallationGuide/html/index.html`
- User's Guide: For Application Developers:
  `http://wwwinfo.cern.ch/asd/geant4/G4UsersDocuments/UsersGuides/ForApplicationDeveloper/html/index.html`
- User's Guide: For Toolkit Developers:
  `http://wwwinfo.cern.ch/asd/geant4/G4UsersDocuments/UsersGuides/ForToolkitDeveloper/html/index.html`
- Software Reference Manual:
  `http://wwwinfo.cern.ch/asdcgi/geant4/SRM/G4GenDoc.csh?flag=1`
- Physics Reference Manual:
  `http://wwwinfo.cern.ch/asd/geant4/G4UsersDocuments/UsersGuides/PhysicsReferenceManual/html/index.html`

4.1.3 FLUKA

FLUKA is an independent, fully integrated particle physics Monte Carlo simulation package. It has many applications in high energy experimental physics and engineering, shielding, detector and telescope design, cosmic ray studies, dosimetry, medical physics, and radio-biology. The software was also originally developed and improved at CERN. Further development is done today only by the *Instituto Nazionale di Fisica Nucleare* in Rome, Italy.

While GEANT3 and GEANT4 are open source software, FLUKA is only available as an executable application. The different detector parts and the logic signal processing have to be defined in a special, less flexible format of input file. For more complex projects it is also possible to link a precompiled FORTRAN77 code, but the possibilities are much more restricted than in GEANT. In order to simplify the detector description with FLUKA, a software to convert GEANT4 detector definitions is under development but not yet available.

The software and further information about the FLUKA project can be found at the official INFN website at `http://www.fluka.org`.

4.2 Response to Galactic Cosmic Rays (SONTEL)

At the outset of this work we decided to use the GEANT4 Monte Carlo code for the simulation of SONTEL. However, during extended tests we encountered serious problems with the low energy hadron-interaction model GHEISHA (Klein and Zoll, 1988) in energy conservation by inelastic hadronic interaction processes. The detailed bug report can be found at

`http://wwwinfo.cern.ch/asdcgi/geant4/problemreport/showbug.cgi?id=359`).

Because of the importance of hadronic processes for this detector the influence of the problem was too big for this model to be used. At that time there was no alternative model available to describe the hadronic processes more accurately as required for SONTEL.

We needed to look for alternative solutions and concluded that GEANT3 provides the best compromise between the functionality of a convenient programming language such as FORTRAN and the possibility of using a well tested model for hadronic interactions such as `GEANT3/FLUKA` for the energy range relevant for SONTEL. Because of the separation of the FLUKA development from CERN, the latest GEANT3 release does not include the most recent FLUKA physics. Nevertheless, GEANT3 allows the study of the implementation of the physical processes and the general structure of the Monte Carlo libraries. Thus, all the simulation applications within this thesis are based on the GEANT3 libraries.

The `SONTEL` GEANT4 application will not be discussed in the following. As soon as an alternative hadronic physics model is available, the existing code can be used for further simulations. The source files and a short description of the command directories are available from the webserver of the Cosmic Ray Group of the University of Bern

`http://kspc4.unibe.ch/sontelgeant.html`.

4.2.1 Introduction

The concept of writing a Monte Carlo application using GEANT3 is straightforward. Most of the routines needed by GEANT3 are predefined by default. In order to implement a specific virtual detector with its individual materials, geometric arrangement, and signal processing, it is only necessary to adapt an existing routine or to write an additional one. These parts are then compiled, together with the precompiled GEANT3 libraries, to get an executable application. The GEANT3 user manual (CERN, 1993) gives detailed information about each procedure and how they have to be used within the source code.

The first step of the detector simulation includes the determination of the energy and direction dependent efficiency functions and the simulation of the various counting rates of the SONTEL detector. Figure 4.1 schematically shows the sequence of the simulation application.

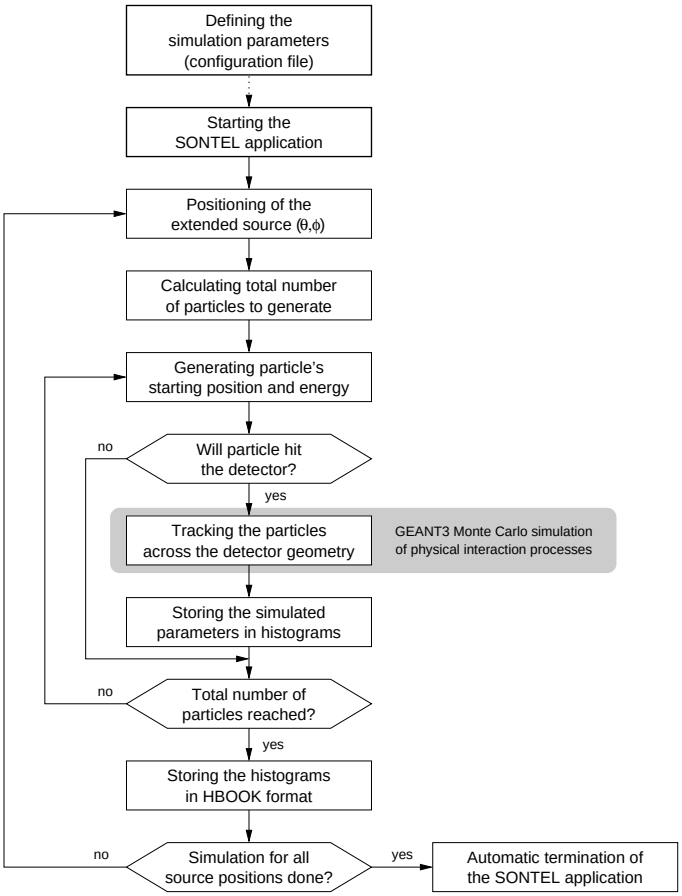


Figure 4.1: Flux diagram of the SONTEL detector simulation application.

The following sections describe the structure of the single program parts and explain the handling of the application.

4.2.2 Detector Geometry

A major part of the source code consists of the implementation of the exact detector geometry and material. The description of the virtual detector has to be included within the user routine UGEOM which is listed in detail in Appendix A.1.10.

In a first step all the used materials are defined. As discussed in Chapter 2 the characteristic parameters such as atomic composition, density, and temperature for most materials are known. The unknown parameters are estimated. The different parts of the detector are then defined as geometrical objects consisting of one of the defined materials. The whole detector is modelled by copying and arranging the parts within the virtual laboratory environment. The sensitive parts of the virtual detector, i.e. the proportional counters and the scintillators, in which information about the physical processes is registered, are also defined within this user routine.

Figures 4.2 and 4.3 show two different views of the GEANT3 implemented detector geometry and also provide a look inside the detector.

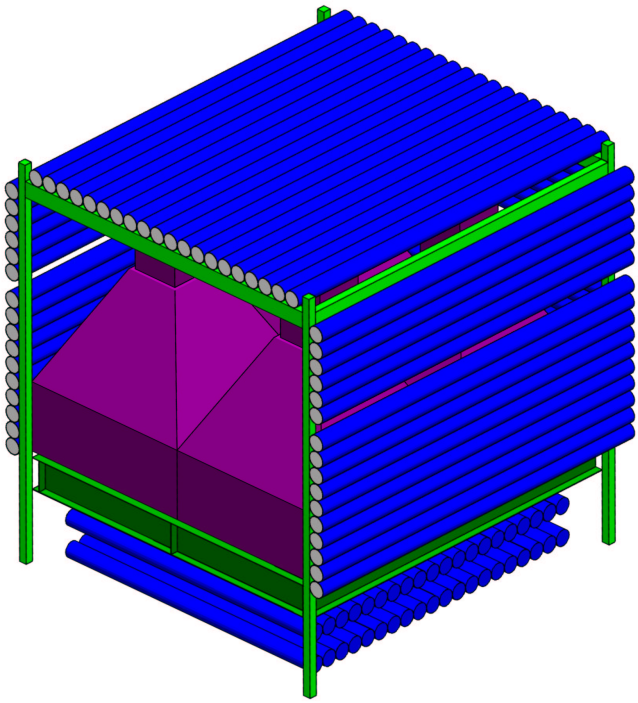


Figure 4.2: View of the virtual SONTEL detector. The left hand side consisting of proportional counters is removed in order to show the inside.

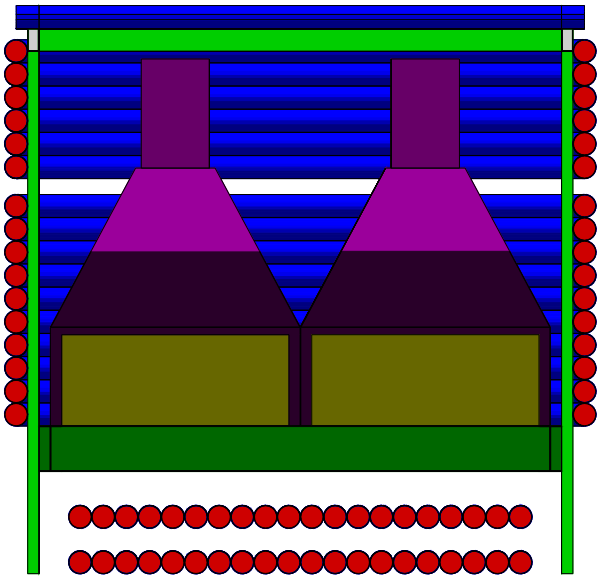


Figure 4.3: Frontal cut through the modeled SONTEL detector. Different colours indicate the different parts of the detector: Metallic proportional counter tubes (blue), proportional counter gas filling (red), metallic scintillator cover (magenta), scintillator blocks (olive-green) and metallic parts of the rack (green).

4.2.3 Particle Generation

In the SONTEL Monte Carlo application there are two different modes of particle sources. Figure 4.4 illustrates the two possibilities.

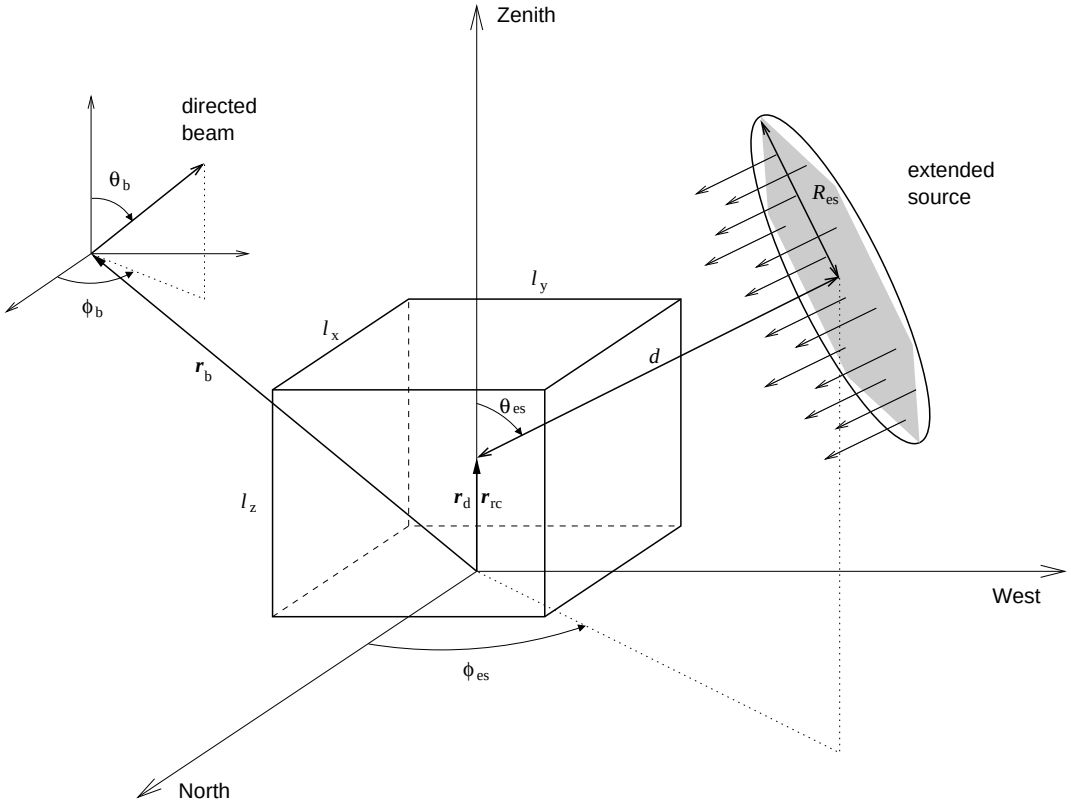


Figure 4.4: Arrangement of the two available types of particle sources of the SONTEL application. For description of the parameters see text.

One possibility is to define a point-like source at the position $\boldsymbol{r}_\mathrm{b}$. This source (*directed beam*) emits particles of a certain type towards the direction characterized by the zenith angle θ_b and the azimuth angle ϕ_b. This type of source can only be used in the interactive application (see Section 4.2.7).

For more realistic simulations an *extended source* must be used. The extended source is a circular plane with radius R_es emitting particles parallel to its surface normal. Usually the center of rotation $\boldsymbol{r}_\mathrm{rc}$ is set equal to the center of the detector $\boldsymbol{r}_\mathrm{d}$. The position of the source is defined by the position of the rotation center $\boldsymbol{r}_\mathrm{rc}$, the distance d from the rotation center to the center of the source, the zenith angle θ_es and the azimuth angle ϕ_es of rotation. In order to emit only the particles crossing the detector volume, an intersection check can be enabled. Thus, only particles originating from a certain area of the circular plane are tracked (grey shadowed area in Figure 4.4).

The zenith angle θ_es and the azimuth angle ϕ_es do not have to be defined when using the application in batch mode. Due to the symmetry of the detector the calculations have to be made for one octant only. According to the number of desired zenith angles N_θ and azimuth angles N_ϕ within one octant, the position of the extended source is set by the application itself. The position angles are calculated to receive simulation results for equal

solid angle elements $|\Delta\Omega| = |\Delta \cos\theta \cdot \Delta\phi|$. Accordingly, the zenith angles are given by

$$\theta_i = \frac{\pi}{2} - \arccos\left(\frac{i}{N_\theta - 1}\right) \quad (i = 0 \ldots N_\theta - 1) \tag{4.1}$$

and the azimuth angles by

$$\phi_j = \frac{j}{N_\theta - 1} \cdot \frac{\pi}{2} \quad (j = 0 \ldots N_\phi - 1). \tag{4.2}$$

In order to simulate the galactic cosmic ray background the integral intensity from the zenith direction is set to $I_0 = I(\theta = 0°)$ and the exponent of the $\cos^n$ law to n_p. The intensity $I(\theta_\mathrm{es})$ of emitted particles from the source at zenith angle θ_es is calculated to

$$I(\theta_\mathrm{es}) = I_0 \, \cos^{n_\mathrm{p}} \theta_\mathrm{es}. \tag{4.3}$$

Furthermore, the total number of generated particles N_tot for one simulation run depends on the sampling time t and the area of the source A_es. If the test of intersection is enabled, only a certain fraction N_eff of the generated particles is tracked depending on the position of the source, i.e.

$$N_\mathrm{tot} = I(\theta_\mathrm{es}) \cdot A_\mathrm{es} \cdot \Delta\Omega \cdot t = I_0 \cdot \cos^{n_\mathrm{p}} \theta_\mathrm{es} \cdot \pi R_\mathrm{es}^2 \cdot \frac{1}{N_\theta - 1} \frac{\pi/2}{N_\phi - 1} \cdot t \geq N_\mathrm{eff}. \tag{4.4}$$

The last parameter, the kinetic energy T of the generated particles, can be assigned in three different ways. The energy can be set to be

- monoenergetic: $T = \text{const.}$,
- uniformly distributed: $T_\mathrm{min} \leq T \leq T_\mathrm{max}$,
- arbitrarily distributed according to an implemented differential spectrum.

In order to simulate the cosmic ray background at Gornergrat level, the differential energy spectra for each particle type are implemented according to the data presented in Chapter 3.

The user routine `GUKINE` contains the full implementation of the extended particle source and the generator of the energy spectra. For details of the implementation of these topics see the corresponding source code in Appendix A.1.4.

4.2.4 Physical Interaction Processes

The simulation of the particle's interaction processes within the traversed medium starts after the generation of a particle emission. The trajectories of the primary and secondary particles are divided into small steps. The individual step size is calculated by the Monte Carlo method according to the involved cross sections, respecting boundary values as defined manually in the configuration file (see Section 4.2.7).

The energy deposited in all steps within one of the sensitive detector parts is summed up and recorded by the routine `GUSTEP` listed in Appendix A.1.7.

GEANT3 is able to handle many different physical interaction processes. For the SONTEL simulation, the following processes were activated:

- Bremsstrahlung
- Compton scattering
- electron-positron annihilation
- hadronic processes
- energy loss by ionisation
- multiple scattering
- muon nuclear interaction
- pair production
- particle decay
- photoelectric effect

Each of the listed process types is represented by one or several physics models. As already mentioned, an important reason why GEANT3 is used in this work is the availability of the hadronic physics of the FLUKA code, a complete alternative to the `GHEISHA` model. The detector application uses the `GEANT3/FLUKA` interface for hadron energies above 20 MeV and the `GEANT3/MICAP` interface (Monte Carlo Ionization Chamber Analysis Package) for lower energies. The corresponding declarations are included within the user routines `GUHADR` and `GUPHAD` listed in Appendices A.1.3 and A.1.6.

For further details on the listed physical processes we refer to the corresponding chapters of the GEANT3 user manual (CERN, 1993).

4.2.5 Particle Detection

At the end of a run simulating all interaction processes after one particle emission, the routine `GUOUT` (Appendix A.1.5) is called. This routine implements the logic data acquisition of the detector as described in Chapter 2.

The veto signal is obtained from the energy deposit within the proportional counters. If the deposited energy E_{dep} within at least one counter tube n is greater than the minimum energy to produce an electron-ion-pair, i.e.

$$E_{\mathrm{dep},n} \geq E_{\mathrm{min}} \simeq 26.2\,\mathrm{eV}, \tag{4.5}$$

a veto pulse is generated. According to Allkofer (1971) it can be assumed that a single electron is enough to trigger a counting pulse. The information whether a veto signal is generated or not is recorded.

The directional signals are generated similarly. To obtain a pulse from a directional counter tube, condition (4.5) has to be fulfilled. Then the pulses of the proportional counters in the upper and in the lower level are compared for pairwise coincidences as illustrated in Figure 2.4. If any of the directional conditions are fulfilled, the corresponding information is recorded, together with the number of conditions fulfilled (*multiplicity*).

In order to obtain a scintillator pulse the deposited energy within each scintillator is used. A counting signal from an energy channel ch1 to ch4 is registered if at least one scintillator unit k fulfills the condition

$$\begin{aligned} \text{ch1} &\Leftrightarrow E_{\text{dep},k} > 40\,\text{MeV}, \\ \text{ch2} &\Leftrightarrow E_{\text{dep},k} > 80\,\text{MeV}, \\ \text{ch3} &\Leftrightarrow E_{\text{dep},k} > 120\,\text{MeV}, \\ \text{ch4} &\Leftrightarrow E_{\text{dep},k} > 160\,\text{MeV}. \end{aligned} \tag{4.6}$$

It is assumed that the entire deposited energy within the scintillator units will be detected by the photomultipliers in form of scintillation light. The counting signals of the various energy channels, both with and without anticoincidence of the veto signal, are stored. Furthermore, a directional signal "with anti" is generated if the corresponding pure directional signal coincides with a scintillator signal "ch1 with anti". Additionally, the deposited energy within the scintillator material, as well as the initial energy of the particle causing a signal in the scintillation and/or the directional counters, are recorded.

After completion of a simulation process for a fixed particle source and for a specific spectral distribution of source particles, the summed up information on particle detection is available in form of histograms. A complete overview of the output data is given in Section 4.2.8. The definitions of the various histograms are included in the user routine `UHINIT` listed in Appendix A.1.13.

4.2.6 Installation

Hardware and Software Prerequisites

The `SONTEL` GEANT3 application is intended for use with the CERN Program Library (CERNLIB). It was developed with the release 3.21 of GEANT under a Linux platform. CERNLIB is supported by the following operating systems:

- Most flavors of UNIX from vendors Sun, DEC, HP, SGI,
- Linux on PC with gcc and g77 compiler,
- Microsoft Windows NT/95/98/2000/XP with Microsoft Visual C/C++ and Digital Visual Fortran.

More information about the installation of the CERNLIB distribution can be found at
`http://wwwinfo.cern.ch/asd/cernlib/version.html`.

Source Files and Directory Structure

The FORTRAN77 source code of the `SONTEL` GEANT3 application is available in the tar-gzip file `SONTEL.tar.gz`. In the directory containing the file `SONTEL.tar.gz`, the command

```
tar -xvzf SONTEL.tar.gz
```

produces the directory `./SONTEL` with one sub-directory `./SONTEL/hbook`. The main directory contains the source code with all its corresponding files and also the data processing

routines for use with *Physics Analysis Workstation (PAW)*. The sub-directory is used to store the simulation output.

After extracting `sontel.tar.gz` it is essential to verify that the aliases `xneut95.dat` and `flukaaf.dat` point to the corresponding files of the CERNLIB distribution, which is usually located within the directory `./XXXX/src/geant321/data/`, where `XXXX` is the year of the release.

Table 4.1 lists the files contained in `SONTEL.tar.gz` and indicates their function.

`.forli`	Script to compile `sontel` (see Section 4.2.7)
`.forli2`	Script to compile `sontelb` (see Section 4.2.7)
`.forli3`	Script to compile `sonteleff` (see Section 4.2.7)
`celoss.inc`	FORTRAN77 include-file (Appendix A.1.14)
`flukaaf.dat`	Alias to FLUKA data file
`flukaerr.dat`	FLUKA error messages
`guhadr.F`	Implements hadronic physics (Appendix A.1.3)
`gukine.F`	Generates kinetics for primary particles (Appendix A.1.4)
`guout.F`	Saves parameters at the end of each event (Appendix A.1.5)
`guphad.F`	Implements hadronic physics (Appendix A.1.6)
`gustep.F`	Saves parameters at the end of each tracking step (Appendix A.1.7)
`gutrev.F`	Controls tracking of one event (Appendix A.1.8)
`main.F`	Main program of batch application `sontelb` (Appendix A.1.1)
`main2.F`	Main program of batch application `sonteleff` (Appendix A.1.2)
`pvolum.inc`	FORTRAN77 include-file (Appendix A.1.15)
`sontel`	Interactive GEANT3 application
`sontel.ffk`	Configuration file for applications (Section 4.2.7)
`sontel.inc`	FORTRAN77 include-file (Appendix A.1.16)
`sontelb`	Simulates the detector for one specific particle flux spectrum
`sonteleff`	Simulates detection efficiencies for a flat particle flux spectrum
`ufiles.F`	Handles input and output files (Appendix A.1.9)
`ugeom.F`	Contains the geometrical implementation of the detector (Appendix A.1.10)
`uginit.F`	Reads the configuration file and initializes the spectra (Appendix A.1.11)
`uglast.F`	Prints histograms and statistics (Appendix A.1.12)
`uhinit.F`	Defines the used histograms (Appendix A.1.13)
`xsneut95.dat`	Alias to neutron cross section data

Table 4.1: Files of the GEANT3 `SONTEL` application and their function.

Compilation

Provided the CERNLIB is set up correctly, it should only be necessary to enter the commands `.forli`, `.forli2` and `.forli3` within the directory `./SONTEL`. These commands will build up the needed libraries and the corresponding executables in the same directory.

4.2.7 Configuration and Execution

Environment Variables

For the SONTEL application only the following environment variables have to be set up (definitions may differ for individual GEANT3 installations):

```
export CERN=~/G3
export CERN_LEVEL=2000
export CERNLIB_DOUBLE=1
```

Configuration parameters have to be declared within the separate configuration file.

Configuration File

The configuration file `sontel.ffk` contains the essential parameters to configure the simulation process. Table 4.2 summarizes the parameters used to control the individual simulation process, while Table 4.3 contains general parameters, e.g. to control native GEANT3 parameters or to define the detector geometry. Once the detector geometry is defined, the latter does not have to be changed except for testing the software.

Execution

The SONTEL GEANT3 application can be started by typing `./sontel`, `./sontelb` or `./sonteleff`.

By typing `./sontel` the interactive version of SONTEL will be started. This executable is mainly used to test the source, the spectrum generator, or the detection modules of the virtual SONTEL detector. For a list of the available commands in the interactive mode and the corresponding explanations see the GEANT3 user manual (CERN, 1993).

The program `sontelb` runs the simulation in the batch mode. The simulation will be executed for each possible direction of incidence given by (4.1) and (4.2). The total number of generated particles for each direction is given by (4.4). At the end of the whole simulation process the program terminates automatically.

In order to determine the energy and direction dependent efficency of the detector the program `sonteleff` is used. The only difference to `sontelb` is that `sonteleff` generates a constant number of particles at all positions of the source.

4.2.8 Output

The simulation results are stored in HBOOK files which can be read and processed by PAW. Each of these HBOOK files contains the following histograms:

- Energy spectrum of the primary particles,
- Position and direction of the generated primary particles,
- Counting rate of the veto counters,
- Counting rate of the photomultipliers with and without anti,

`COSEXP` n_p	Sets the exponent n_p of the $\cos^{n_\mathrm{p}}$ law to calculate the zenith angle dependent cosmic ray intensity.
`FILENR` *nnnnn*	Sets the prefix number *nnnnn* of the output files.
`KINE` i T_1 T_2	Sets primary particle kinetic energy T [GeV] $i = 0$: Monoenergetic, $T = T_1$ $i = 1$: Uniformly distributed, $T_1 \leq T \leq T_2$
`NRPHI` N_ϕ	Sets number N_ϕ of used azimuth angles in one octant.
`NRTHETA` N_θ	Sets number N_θ of used zenith angles in one octant.
`PART` k	Sets primary particle type k $k = 1$: Gamma $k = 3$: Electron $k = 6$: Muon $k = 13$: Neutron $k = 14$: Proton
`PHI` ϕ_b	Sets the azimuth angle ϕ_b [°] of the directed beam.
`POSI` x_b y_b z_b	Sets the position $\boldsymbol{r}_\mathrm{b} = (x_\mathrm{b}, y_\mathrm{b}, z_\mathrm{b})$ [cm] of the directed beam.
`RNDM` N_1 N_2	Sets the initial random number seed.
`SPECTRUM` k	Sets the energy spectrum of the primary particles $k = 0$: Monoenergetic (according to settings made by `KINE`) $k = 1$: Electrons (Section 3.1) $k = 3$: Protons (Section 3.2) $k = 4$: Muons (Section 3.3) $k = 5$: Gammas (Section 3.4) $k = 6$: Neutrons (Section 3.4)
`SPHI` ϕ_es	Sets the azimuth angle ϕ_es [°] of the extended source.
`SRCON` i	Sets type of source $i = 0$: directed beam $i = 1$: extended (planar) source
`STHETA` θ_es	Sets the zenith angle θ_es [°] of the extended source.
`THETA` θ_b	Sets the zenith angle θ_b [°] of the directed beam.
`TIME` t	Sets the sampling time t [s] of the simulation.
`ZENITHINT` I_0	Sets the integral zenith intensity I_0 [$\mathrm{cm^{-2}s^{-1}sr^{-1}}$].

Table 4.2: Configuration parameters in the `sontel.ffk` configuration file.

- Primary energy spectra of particles detected by the photomultipliers with and without anti,
- Total deposit energy in the scintillators,
- Counting rate of the directional counters with and without anti,
- Primary energy spectra of particles detected by the directional counters with and without anti,
- Multiplicity of directional signals.

For each simulated direction of incidence the data are stored within individual files with the following nomenclature: *nnnnn_i_j*`.hbook`. The first number *nnnnn* is the file name as chosen in the configuration file. The second i and the third number j are the indices of the zenith and the azimuth angle as introduced in (4.1) and (4.2).

`ADEEMAX` f_a	Sets the maximum fractional energy loss ($0 < f_\mathrm{a} \leq 1$) of one step in air.
`ADMAXMS` $l_\mathrm{a,max}$	Sets the maximum step length $l_\mathrm{a,max}$ [cm] permitted in air.
`AEPSIL` ϵ_a	Sets the boundary crossing precision ϵ_a [cm] of air.
`ASTMIN` $l_\mathrm{a,min}$	Sets the minimum value $l_\mathrm{a,min}$ [cm] for the maximum step length imposed by energy loss and multiple scattering in air.
`AUTO` i	Disables/enables the automatic calculation of tracking medium parameters ($i = 0, 1$).
`CENTER` x_rc y_rc z_rc	Sets the position $\boldsymbol{r}_\mathrm{rc} = (x_\mathrm{rc}, y_\mathrm{rc}, z_\mathrm{rc})$ [cm] of the rotation center of the source.
`CROSSDET` i	Disables/enables the check function determining if the generated primary hits the detector ($i = 0, 1$).
`DEBUG` i j k	Sets standard GEANT3 debug informations.
`DETCENT` x_d y_d z_d	Sets the position $\boldsymbol{r}_\mathrm{d} = (x_\mathrm{d}, y_\mathrm{d}, z_\mathrm{d})$ [cm] of the center of the detector.
`DETSIZ` l_x l_y l_z	Sets the dimension $l_x \times l_y \times l_z$ [cm] of the detector geometry.
`DISCRI` E_1 E_2 E_3 E_4	Sets the four discrimination levels $E_1 \ldots E_4$ [MeV] of the photomultiplier signal.
`DISTANCE` d	Sets the distance d [cm] of the source from the center of rotation.
`GDEEMAX` f_g	Sets the maximum fractional energy loss ($0 < f_\mathrm{g} \leq 1$) of one step in gas except air.
`GDMAXMS` $l_\mathrm{g,max}$	Sets the maximum step length $l_\mathrm{g,max}$ [cm] permitted in gas except air.
`GEPSIL` ϵ_g	Sets the boundary crossing precision ϵ_g [cm] of gas except air.
`GSTMIN` $l_\mathrm{g,min}$	Sets the minimum value $l_\mathrm{g,min}$ [cm] for the maximum step length imposed by energy loss and multiple scattering in gas except air.
`HADR` i	Disables/enables hadronic physical processes ($i = 0, 1$).
`IONI` E_FIP	Sets the first ionization potential E_FIP [eV] of the counter gas.
`MDEEMAX` f_m	Sets the maximum fractional energy loss ($0 < f_\mathrm{m} \leq 1$) of one step in metals.
`MDMAXMS` $l_\mathrm{m,max}$	Sets the maximum step length $l_\mathrm{m,max}$ [cm] permitted in metals.
`MEPSIL` ϵ_m	Sets the boundary crossing precision ϵ_m [cm] of metals.
`MSTMIN` $l_\mathrm{m,min}$	Sets the minimum value $l_\mathrm{m,min}$ [cm] for the maximum step length imposed by energy loss and multiple scattering in metals.
`PDEEMAX` f_p	Sets the maximum fractional energy loss ($0 < f_\mathrm{p} \leq 1$) of one step in polystyrol.
`PDMAXMS` $l_\mathrm{p,max}$	Sets the maximum step length $l_\mathrm{p,max}$ [cm] permitted in polystyrol.
`PEPSIL` ϵ_p	Sets the boundary crossing precision ϵ_p [cm] of polystyrol.
`PSTMIN` $l_\mathrm{p,min}$	Sets the minimum value $l_\mathrm{p,min}$ [cm] for the maximum step length imposed by energy loss and multiple scattering in polystyrol.
`SORAD` R_es	Sets the radius R_es [cm] of the extended source.
`SONDB` i	Disables/enables SONTEL specific debug informations on the screen ($i = 0, 1$).
`SWITCHPC` i j	Disables/enables the veto counters ($i = 0, 1$) and the directional counters ($j = 0, 1$).
`VDEEMAX` f_v	Sets the maximum fractional energy loss ($0 < f_\mathrm{v} \leq 1$) of one step in vacuum.
`VDMAXMS` $l_\mathrm{v,max}$	Sets the maximum step length $l_\mathrm{v,max}$ [cm] permitted in vacuum.
`VEPSIL` ϵ_v	Sets the boundary crossing precision ϵ_v [cm] of vacuum.
`VSTMIN` $l_\mathrm{v,min}$	Sets the minimum value $l_\mathrm{v,min}$ [cm] for the maximum step length imposed by energy loss and multiple scattering in vacuum.

Table 4.3: General configuration parameters in the `sontel.ffk` configuration file.

4.2.9 Data Processing

Two data processing routines are written for use with PAW. The first one, `write.kumac`, extracts the stored data from the specified `HBOOK` files and writes each data set to an individual ASCII file (`.txt`) with the same prefix number *nnnnn*. The script `integration.kumac` is used to integrate the simulated data over the upper half space of the detector. The data sets contained in the specified ASCII files are integrated numerically and stored in a new ASCII file.

4.3 Secondary Spectra of Solar Particle Events (SOLSPEC)

In order to evaluate the response of the detector to solar energetic particles, the particle spectra at Gornergrat during a Ground Level Enhancement (GLE) have to be simulated first. For this purpose the application `SOLSPEC` has been developed. Using the simulated energy spectra the sensitivity of SONTEL is then determinded by using application `SONTELSEP`.

4.3.1 Introduction

The principle of simulating the passage of radiation through the Earth's atmosphere with GEANT3 is exactly the same as previously described in Section 4.2 about the SONTEL detector. The installation, compilation, and usage procedure is also quite similar; thus, the following sections only explain the specific aspects of this application.

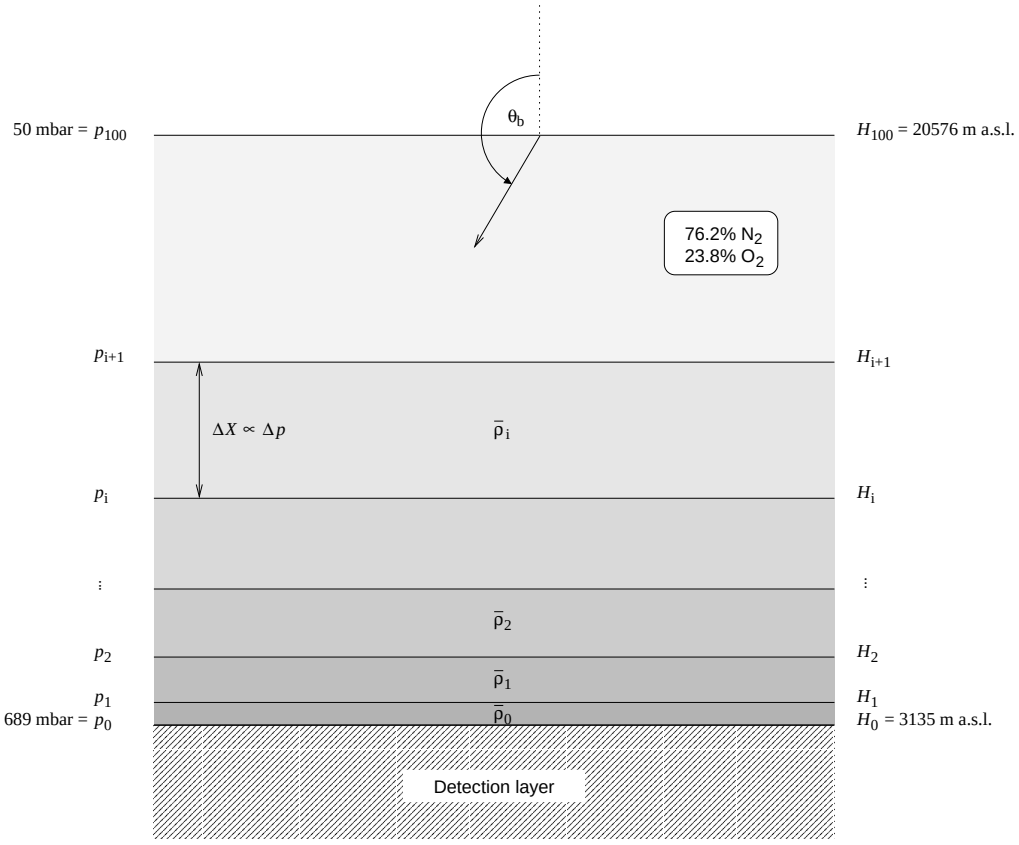


Figure 4.5: Schematic view of the modelled atmosphere. For details see text.

4.3.2 Atmospheric Model

The atmosphere is modelled by stacking 100 parallel layers as illustrated in Figure 4.5. Each of these layers consists of the same chemical composition according to the definition of the U.S. Standard Atmosphere 1976 (Lide, 1998). For simulation purposes, only the two most abundant components, i.e. nitrogen N_2 (76.2% by mass) and oxygen O_2 (23.8% by mass), are considered.

For calculating the thickness ΔH_i and the mean density $\bar{\rho}_i$ of the individual layer i, the subroutines from the FORTRAN77 code `stdatm76.F` from the *University Corporation of Atmospheric Research* (available at `http://dss.ucar.edu/libraries/meteorology/`) are used. The thickness ΔH_i is computed to receive an equal column depth $\Delta X = \bar{\rho}_i \, \Delta H_i$ of each layer. The lower atmospheric boundary is set to Gornergrat level (689 mbar $\hat{=}$ 3135 m a.s.l.). The upper boundary is chosen to 50 mbar ($\hat{=}$ 20.6 km a.s.l.) corresponding to $\sim 95\%$ of the total atmospheric depth.

The different tracking media parameters for the individual layers are calculated by the application itself. The maximum permitted step length $l_{i,\max}$ of layer i varies according to $l_{i,\max} \propto \Delta H_i \propto \bar{\rho}_i^{-1}$ in order to obtain an equal maximum number of steps within each layer.

The entire implementation of the atmosphere can be found within the user routine `UGEOM` listed in Appendix A.2.6.

4.3.3 Particle Generation

In contrast to `SONTEL` only a directed particle beam is used (see Figure 4.4). There are three possibilities to set the energy of the primary particle, i.e. monoenergetic, uniformly distributed within a specified range, or according to an implemented spectrum. The latter is only declared for protons from the April 15, 2001, solar proton event (see Section 5.3.2).

The intensity and the sampling time can be set in the same way as in the `SONTEL` application.

4.3.4 Particle Detection

Below the bottom atmospheric layer at 3135 m a.s.l. there is a detection layer (hatched box in Figure 4.5) consisting of a vacuum. The parameters of each primary and secondary particle entering this layer, i.e. particle type, energy, and direction of incidence, are stored in histograms. After the registration, the corresponding particle is stopped and is not tracked further.

4.3.5 Installation

The installation procedure of `SOLSPEC` is almost the same as described in Paragraph 4.2.6 for `SONTEL`.

The source code of the GEANT3 application is included in a file `SOLSPEC.tar.gz` which can be extracted by typing

```
tar -xvzf SOLSPEC.tar.gz .
```

In addition to SONTEL this archive file contains the file atmos.F including the subroutines for calculating the atmosphere as mentioned in Paragraph 4.3.2.

In order to start the compilation procedure, one only has to type .forli for the interactive application or .forli2 for the application in the batch mode.

4.3.6 Configuration and Execution

Configuration File

The configuration file of SOLSPEC application contains the same keywords as the SONTEL application. Table 4.4 shows the parameters needed to define a simulation procedure.

FILENR *nnnnn*	Sets the prefix number *nnnnn* of the output file.
KINE i T_1 T_2	Sets primary particle kinetic energy T [GeV] $i = 0$: Monoenergetic, $T = T_1$ $i = 1$: Uniformly distributed, $T_1 \leq T \leq T_2$
NRPHI N_ϕ	Sets number N_ϕ of used azimuth angles in one octant.
NRTHETA N_θ	Sets number N_θ of used zenith angles in one octant.
PART k	Sets primary particle type k $k = 1$: Gamma $k = 3$: Electron $k = 6$: Muon $k = 13$: Neutron $k = 14$: Proton
PHI ϕ_b	Sets the azimuth angle ϕ_b [°] of the directed beam.
POSI x_b y_b z_b	Sets the position $\boldsymbol{r}_\mathrm{b} = (x_\mathrm{b}, y_\mathrm{b}, z_\mathrm{b})$ [cm] of the directed beam.
RNDM N_1 N_2	Sets the initial random number seed.
SPECTRUM k	Sets the energy spectrum of the primary particles $k = 0$: Monoenergetic (according to settings made by KINE) $k = 3$: Protons from the April 15, 2001, solar proton event (Lockwood et al., 2002)
SRCON i	Sets type of source (required) $i = 0$: directed beam
THETA θ_b	Sets the zenith angle θ_b [°] of the directed beam.
TIME t	Sets the sampling time t [s] of the simulation.
ZENITHINT I_0	Sets the integral beam intensity J_0 [$\mathrm{cm}^{-2}\mathrm{s}^{-1}$].

Table 4.4: Configuration parameters in the solspec.ffk configuration file.

Execution

The SOLSPEC GEANT3 application is started by typing either ./solspec or ./solspecb.

By typing ./solspec the interactive version of SOLSPEC is started. This executable is mainly used to test the atmospheric implementation but is also able to execute a complete atmospheric simulation. However, the storage of the produced histograms has to be done manually. For an explanation of the available commands we refer to the GEANT3 user manual (CERN, 1993).

The batch mode application solspecb executes the simulation according to the parameters set in the configuration file and finally stores the resulting data itself.

4.3.7 Output

The resulting histograms of a simulation are stored in a single `HBOOK` file. This file contains the following histograms:

- Energy spectrum of the primary particles,
- Position and direction of the primary particle beam,
- Number of resulting secondary particles for each type in the detection layer,
- Omnidirectional energy spectra of the different particle types in the detection layer,
- Directional energy spectra of the different particle types in the detection layer for the defined number of directions $N = N_\theta \times [4(N_\phi - 1) + 1]$.

4.3.8 Data Processing

In order to reformat the output of `SOLSPEC` for input use by `SONTELSEP` the PAW macro file `writesol.kumac` has been developed. This macro extracts the data from the chosen `HBOOK` file and writes the energy spectrum for each particle type and each direction into an ASCII file (`.txt`). The total number of detected particles of a certain type from each direction is stored in a separate ASCII file (`.int`).

4.4 Response to Solar Cosmic Rays (`SONTELSEP`)

The `SONTELSEP` application is a direct derivate of the `SONTEL` application. The only difference between the two versions is that `SONTELSEP` uses the spectra produced by `SOLSPEC` instead of the implemented secondary spectra. Therefore, the directory of `SONTELSEP` contains an additional subdirectory `./SONTELSEP/spec`. Before starting the simulation the preprocessed `SOLSPEC` output files (see Section 4.3.8) have to be copied to this subdirectory. Table 4.5 lists the available configuration parameters for the `SONTELSEP` application.

`FILENR` *nnnnn*	Sets the prefix number *nnnnn* of the output file.
`NRPHI` N_ϕ	Sets number N_ϕ of used azimuth angles in one octant.
`NRTHETA` N_θ	Sets number N_θ of used zenith angles in one octant.
`PART` k	Sets primary particle type k
	$k = 1$: Gamma
	$k = 3$: Electron
	$k = 6$: Muon
	$k = 13$: Neutron
	$k = 14$: Proton
`RNDM` N_1 N_2	Sets the initial random number seed.
`SPECTRUM` k	Sets the energy spectrum of the primary particles (required)
	$k = 1$: External spectra from the subdirectory `../spec`
`SRCON` i	Sets type of source (required)
	$i = 1$: extended source
`TIME` t	Sets the sampling time t [s] of the simulation.

Table 4.5: Configuration parameters in the `sontelsep.ffk` configuration file.

Chapter 5

Simulation Results

5.1 Detection Efficiencies

In a first step the sensitivity of the various detection channels to the relevant species of particles from different directions was simulated by the `SONTEL` GEANT3 application.

Because the atmospheric cascade, caused by galactic cosmic rays, is not included in this simulation, the correlation between a primary cosmic ray particle and its detection probability at ground level cannot be determined. Thus, within this work it is not possible to derive the *yield function* which is normally used to characterize cosmic ray detectors. Instead, we present the *directional detection efficiency* $\varepsilon_i^n(T, \theta, \phi)$ defined as the ratio between the number of counts N^n in a detection channel n and the total number $N_i(T, \theta, \phi)$ of particles of type i with kinetic energy T hitting the detector from a direction given by the zenith angle θ and the azimuth angle ϕ:

$$\varepsilon_i^n(T, \theta, \phi) = \frac{N^n}{N_i(T, \theta, \phi)}. \tag{5.1}$$

Furthermore, we present the *mean detection efficiency* $\bar{\varepsilon}_i^n(T)$ of a channel n independent of the direction of the radiation. It is defined as the mean of the directional detection efficiency for the upper half space, since no backscattered radiation from the ground is considered:

$$\bar{\varepsilon}_i^n(T) = \frac{1}{2\pi} \int_{\cap} \varepsilon_i^n(T, \theta, \phi)\, \mathrm{d}\Omega. \tag{5.2}$$

In the next two sections we compile the simulation results for the scintillator channels as well as for the directional counters.

5.1.1 Scintillators

Figure 5.1 illustrates the energy dependent directional detection efficiencies for neutrons, protons, muons, electrons (i.e. the four most abundant species of particles in the secondary cosmic radiation), and for γ-rays. As an example the two plots show the efficiencies of the lowest energy channel (> 40 MeV) without and with anti, respectively, for radiation from the zenith direction.

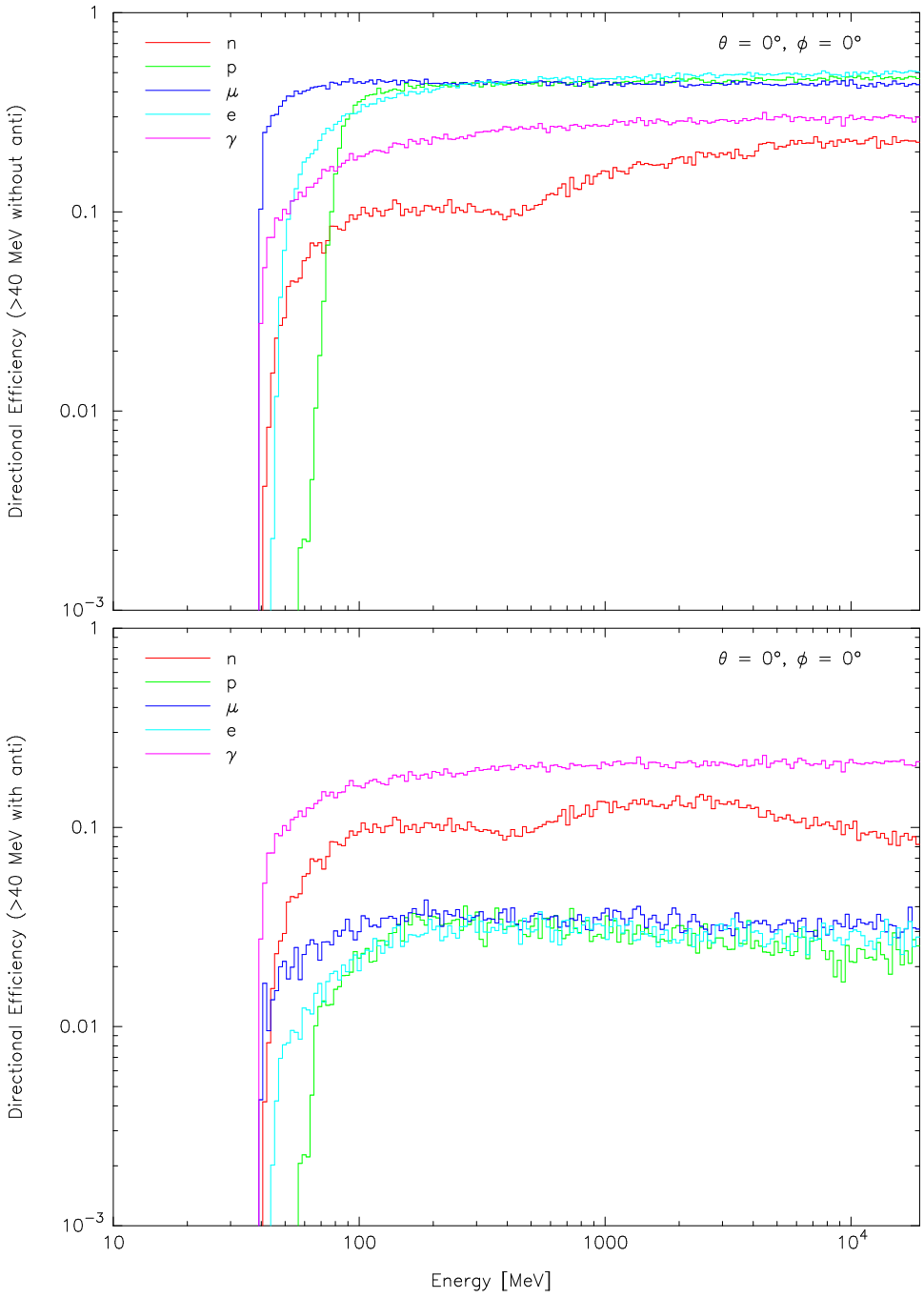


Figure 5.1: Simulated vertical directional efficiencies for neutrons, protons, muons, electrons, and γ-rays. The upper plot refers to the lowest energy channel (> 40 MeV) without anti, the lower plot to the same energy channel with anti.

In consideration of the main purpose of the SONTEL detector, the efficiency functions of the "neutron channels", i.e. of the energy channels with anti, are of special interest. Figure 5.2 shows a comparison of the directional efficiencies of all the energy channels with and without anti. They were simulated for neutrons entering from the zenith direction. The dotted lines indicate the threshold energies of the four detection channels.

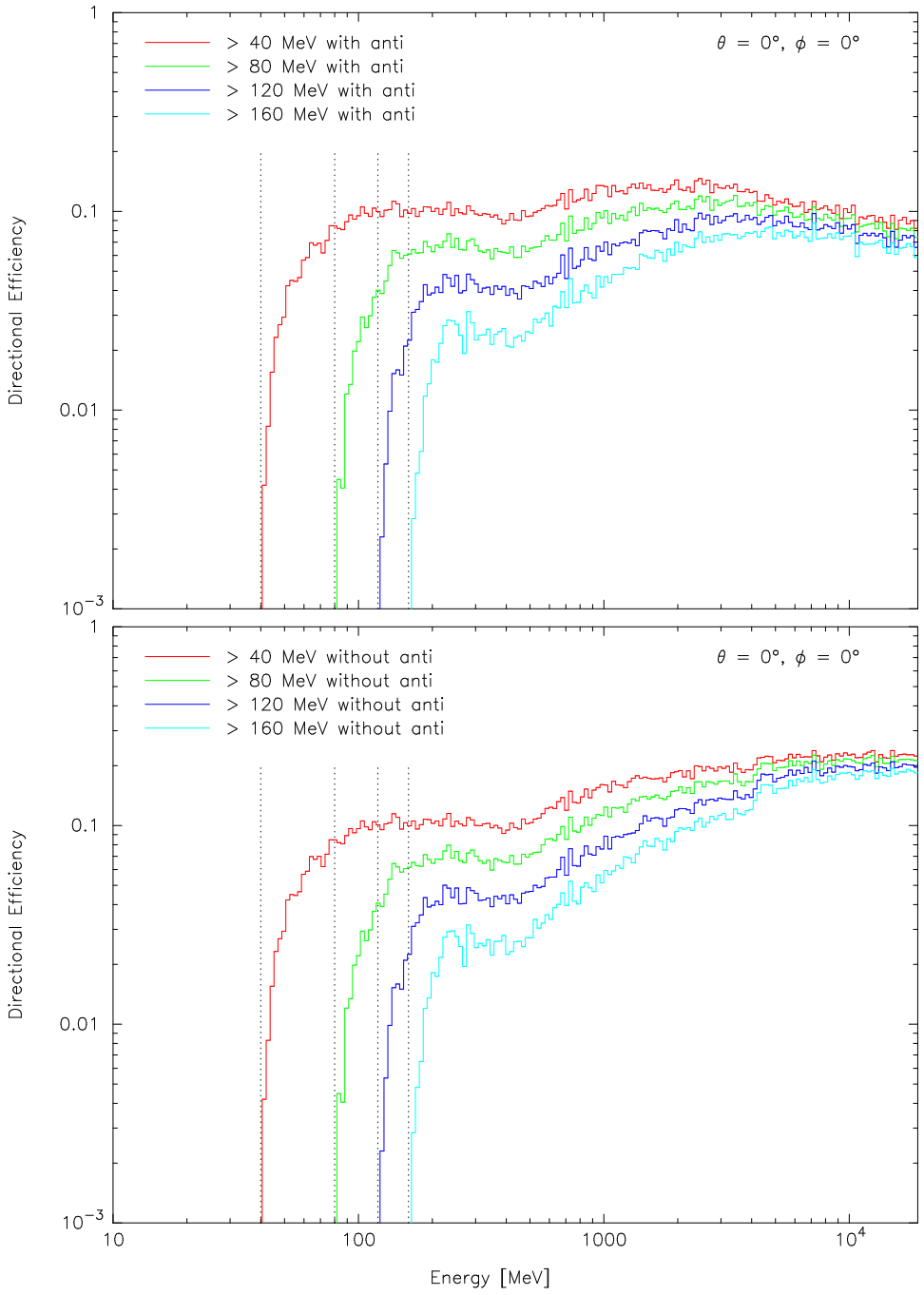


Figure 5.2: Simulated vertical directional efficiencies for neutrons. The upper plot shows the directional efficiencies of the the four energy channels with anti, the lower plot of the energy channels without anti.

In the following we present the directional efficiencies of the four energy channels with and without anti (Figures 5.3 to 5.12) for five species of radiation. Each figure includes results for six directions of incidence with zenith angles increasing from 0° to 90° in steps with $|\Delta \cos\theta| = 0.2$ while the azimuth angle is fixed southwards. Because of the almost symmetric geometry of the detector, the dependence on the azimuth angle is only marginal and therefore the corresponding pictures are not shown.

Figures 5.13 to 5.17 show the mean detection efficiencies of the various scintillator channels for the five radiation types considered.

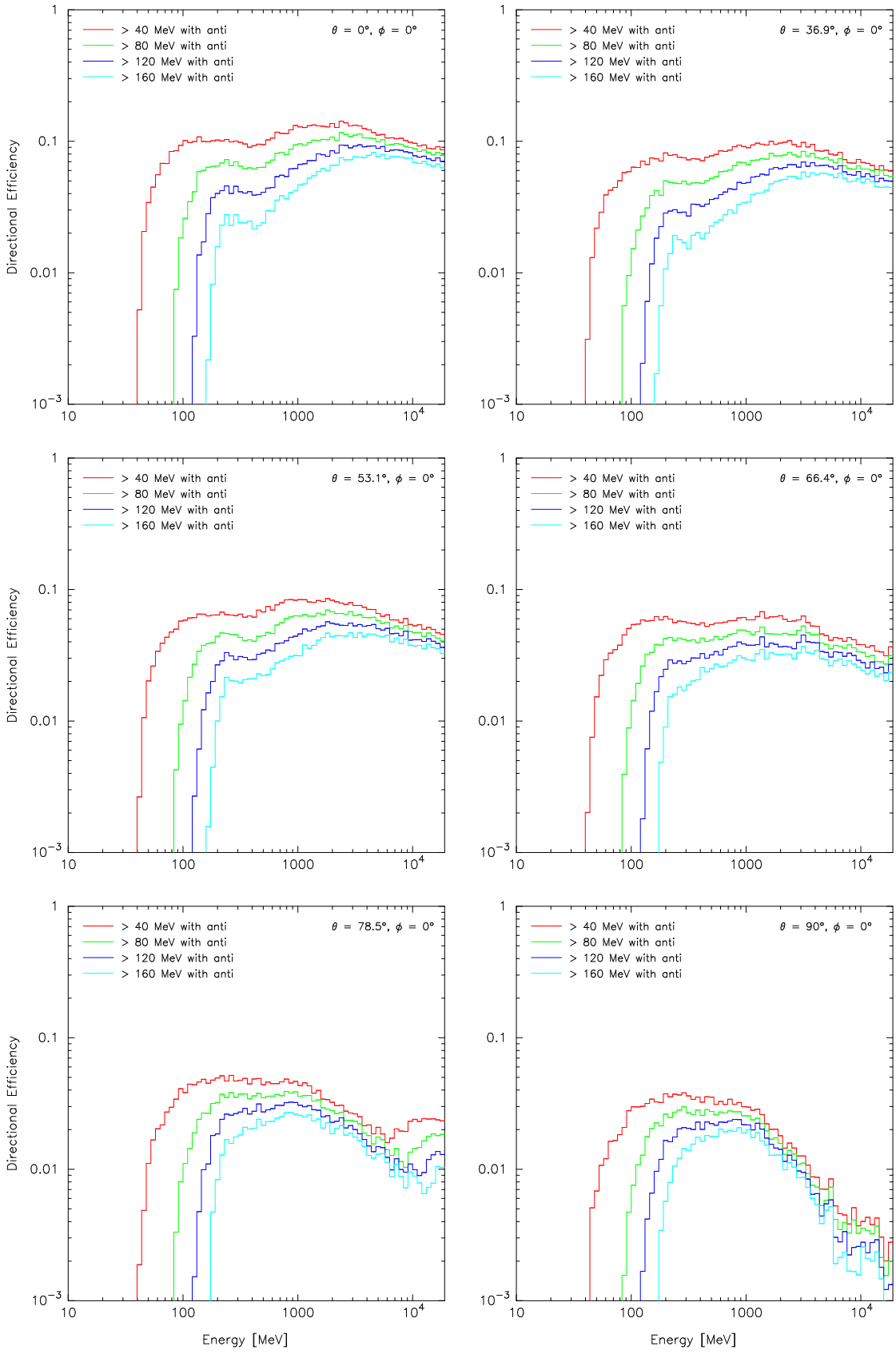


Figure 5.3: Simulated directional efficiencies for neutrons. The plots show the directional efficiencies of the scintillator channels with anti.

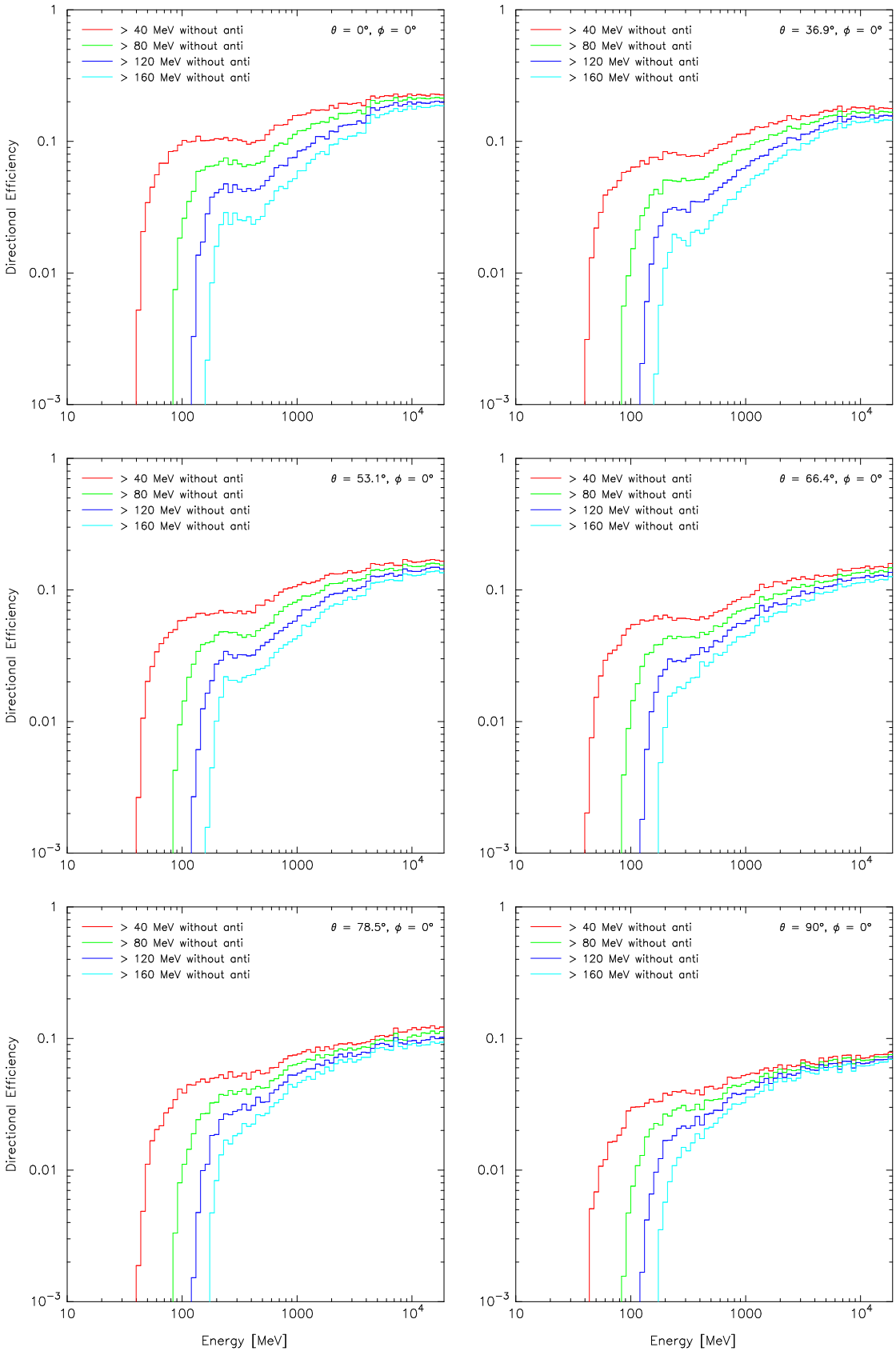


Figure 5.4: Simulated directional efficiencies for neutrons. The plots show the directional efficiencies of the scintillator channels without anti.

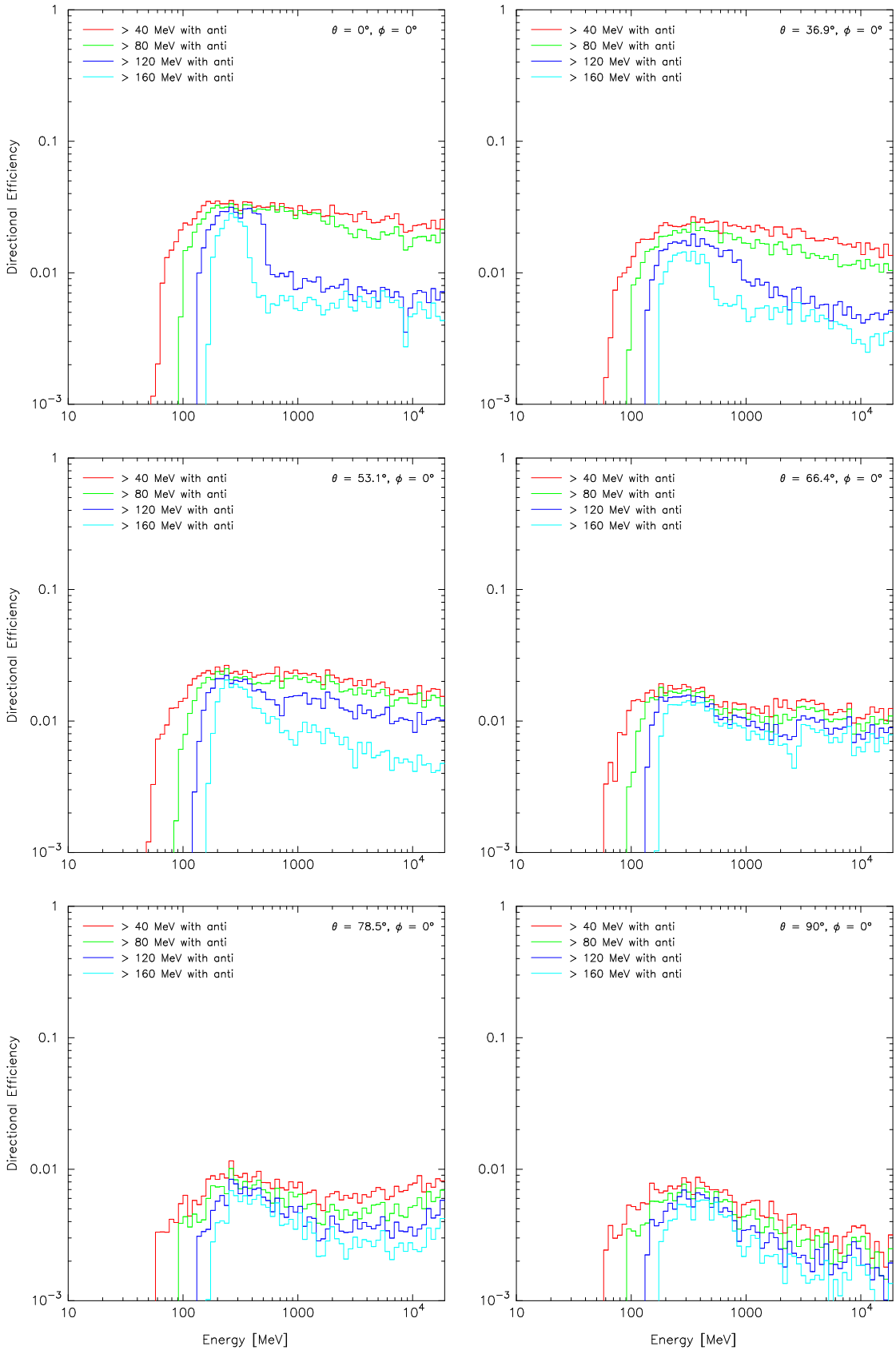


Figure 5.5: Simulated directional efficiencies for protons. The plots show the directional efficiencies of the scintillator channels with anti.

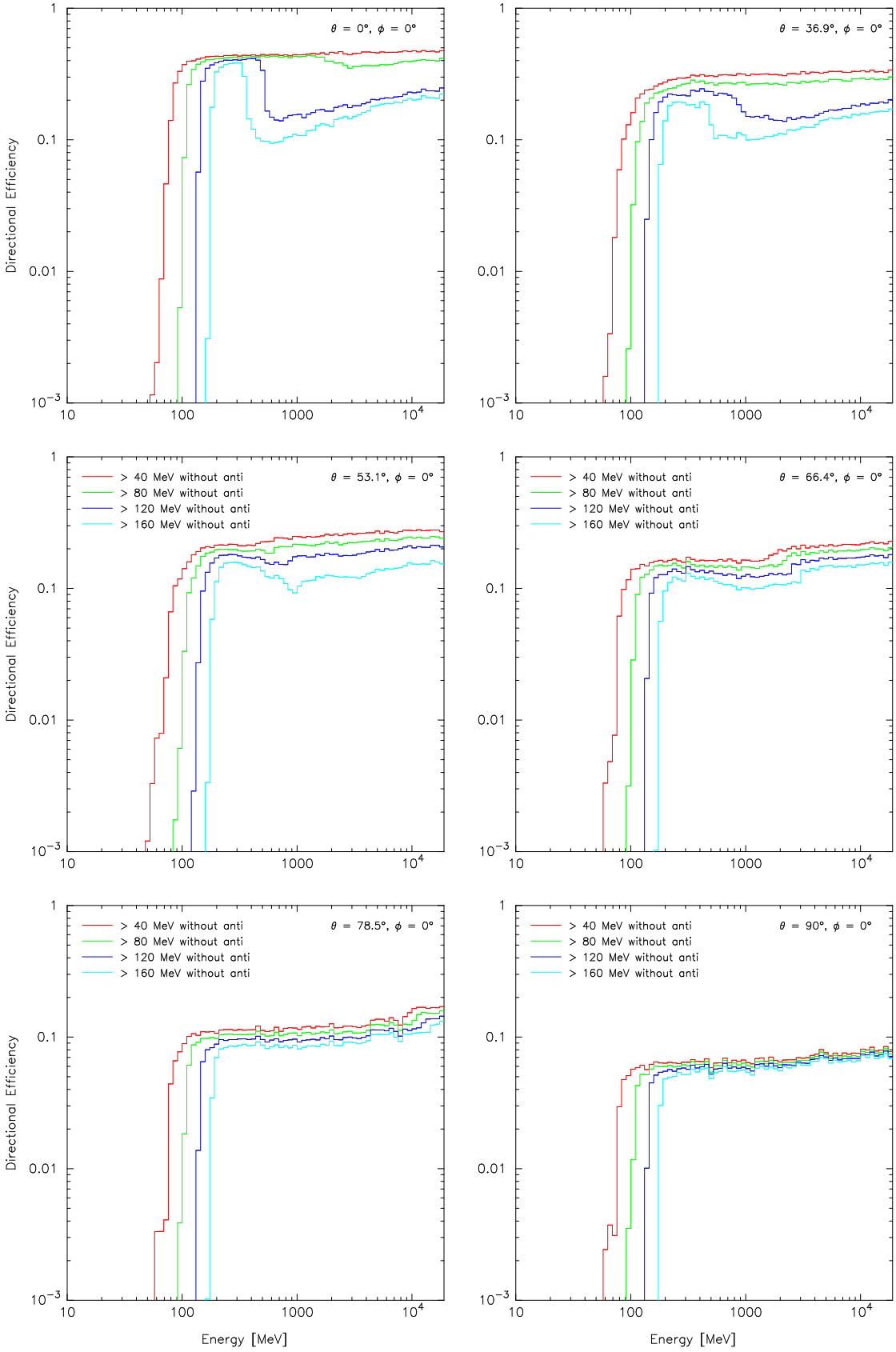


Figure 5.6: Simulated directional efficiencies for protons. The plots show the directional efficiencies of the scintillator channels without anti.

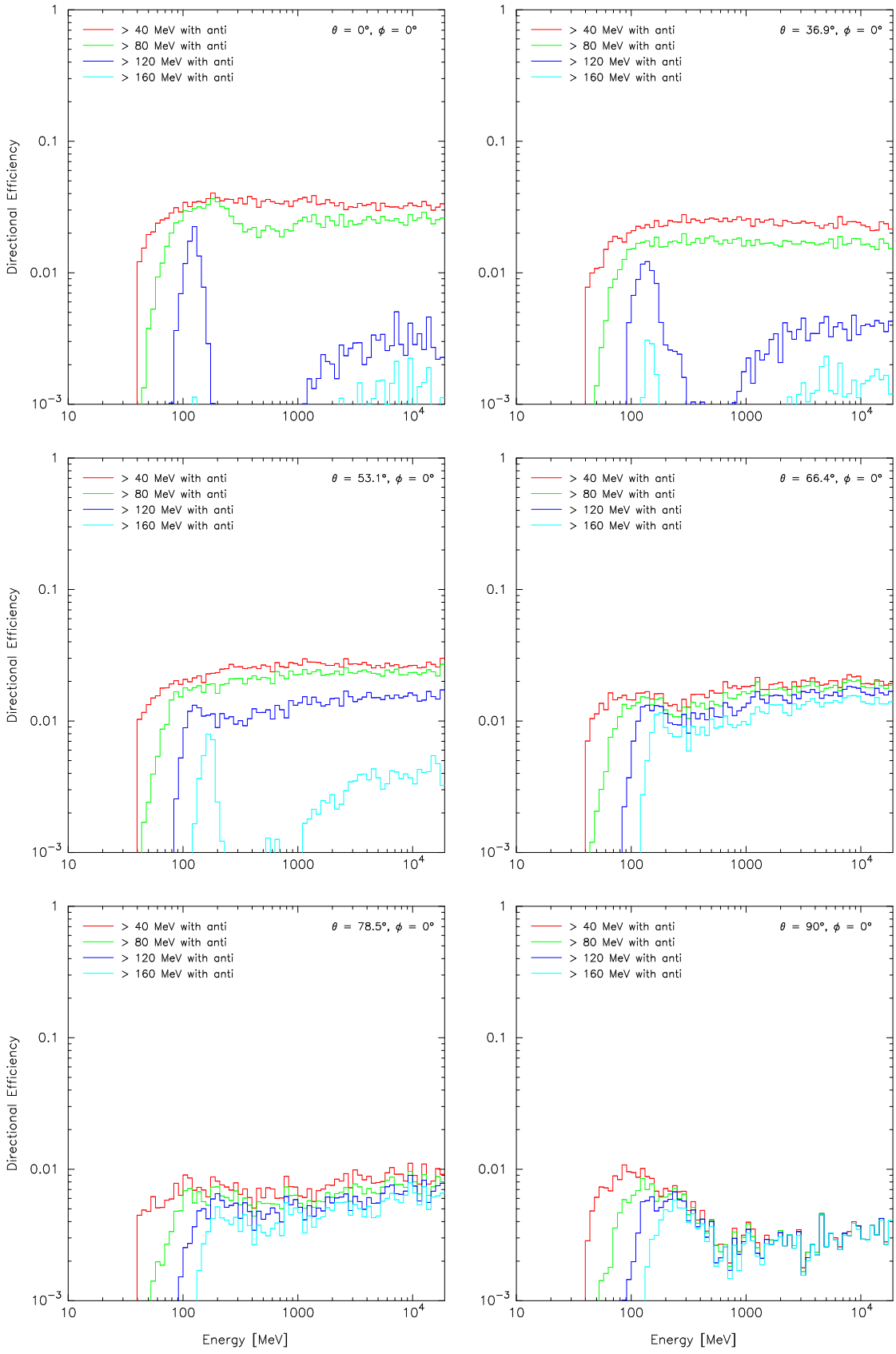


Figure 5.7: Simulated directional efficiencies for muons. The plots show the directional efficiencies of the scintillator channels with anti.

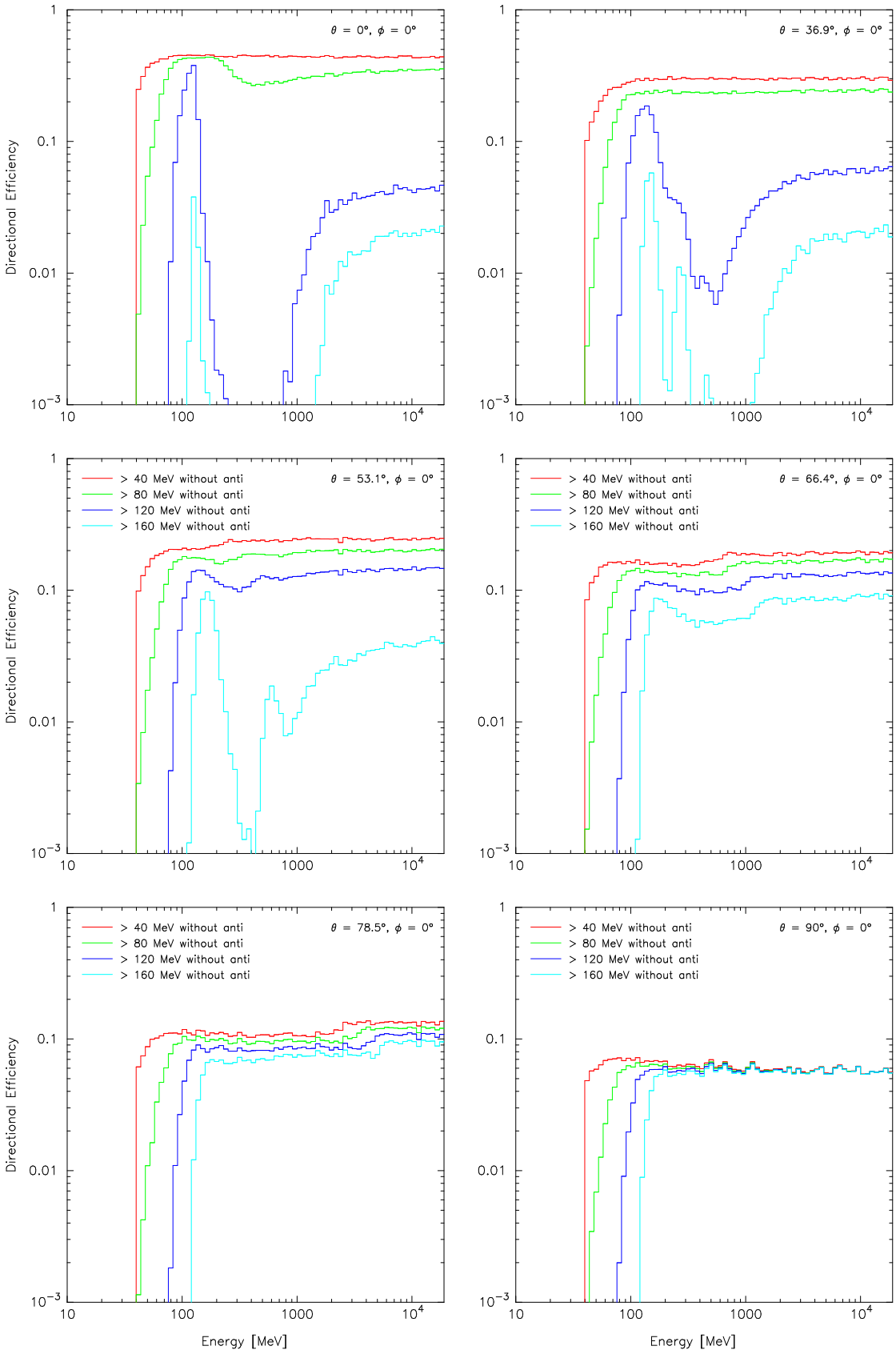


Figure 5.8: Simulated directional efficiencies for muons. The plots show the directional efficiencies of the scintillator channels without anti.

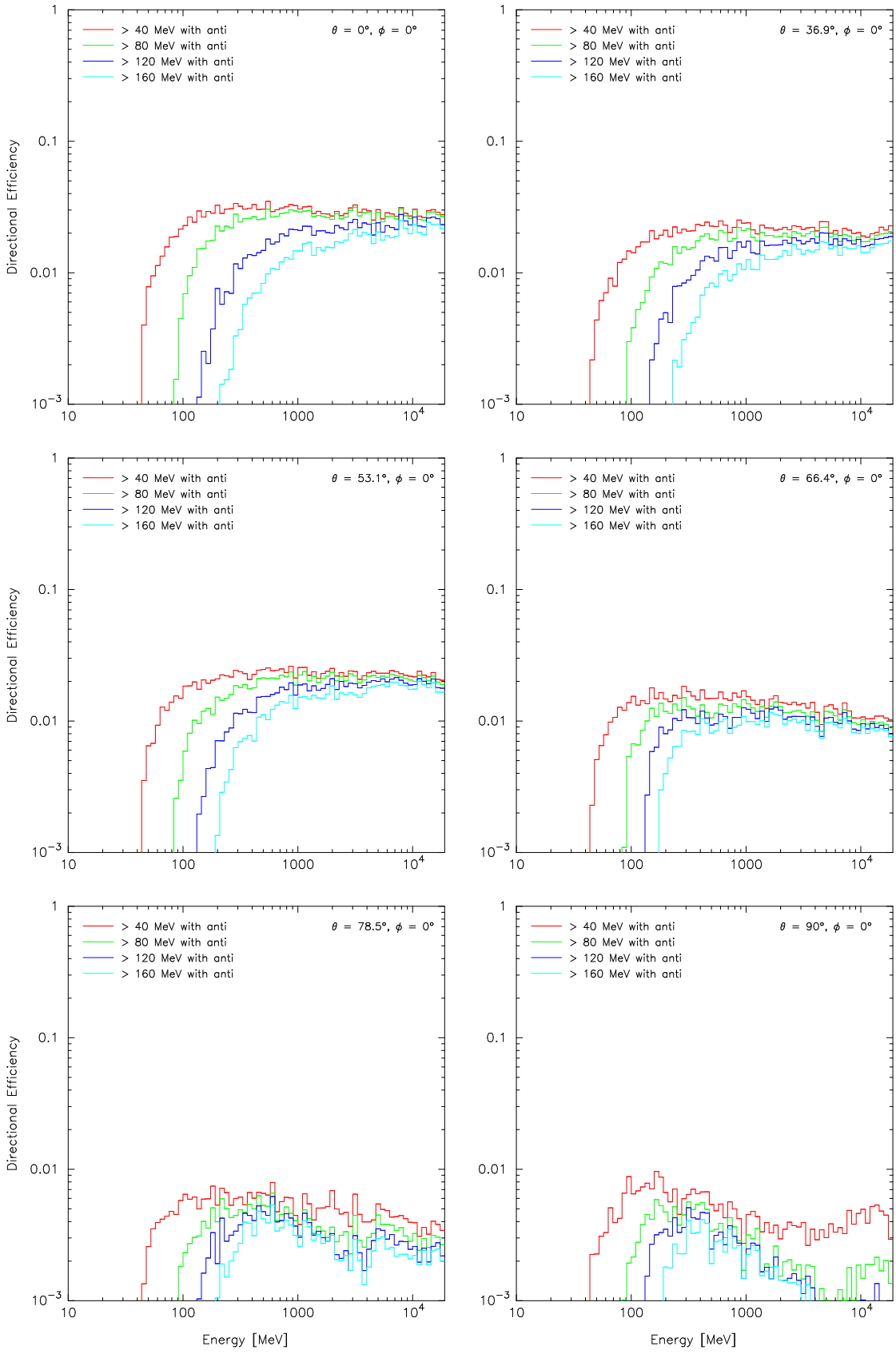


Figure 5.9: Simulated directional efficiencies for electrons. The plots show the directional efficiencies of the scintillator channels with anti.

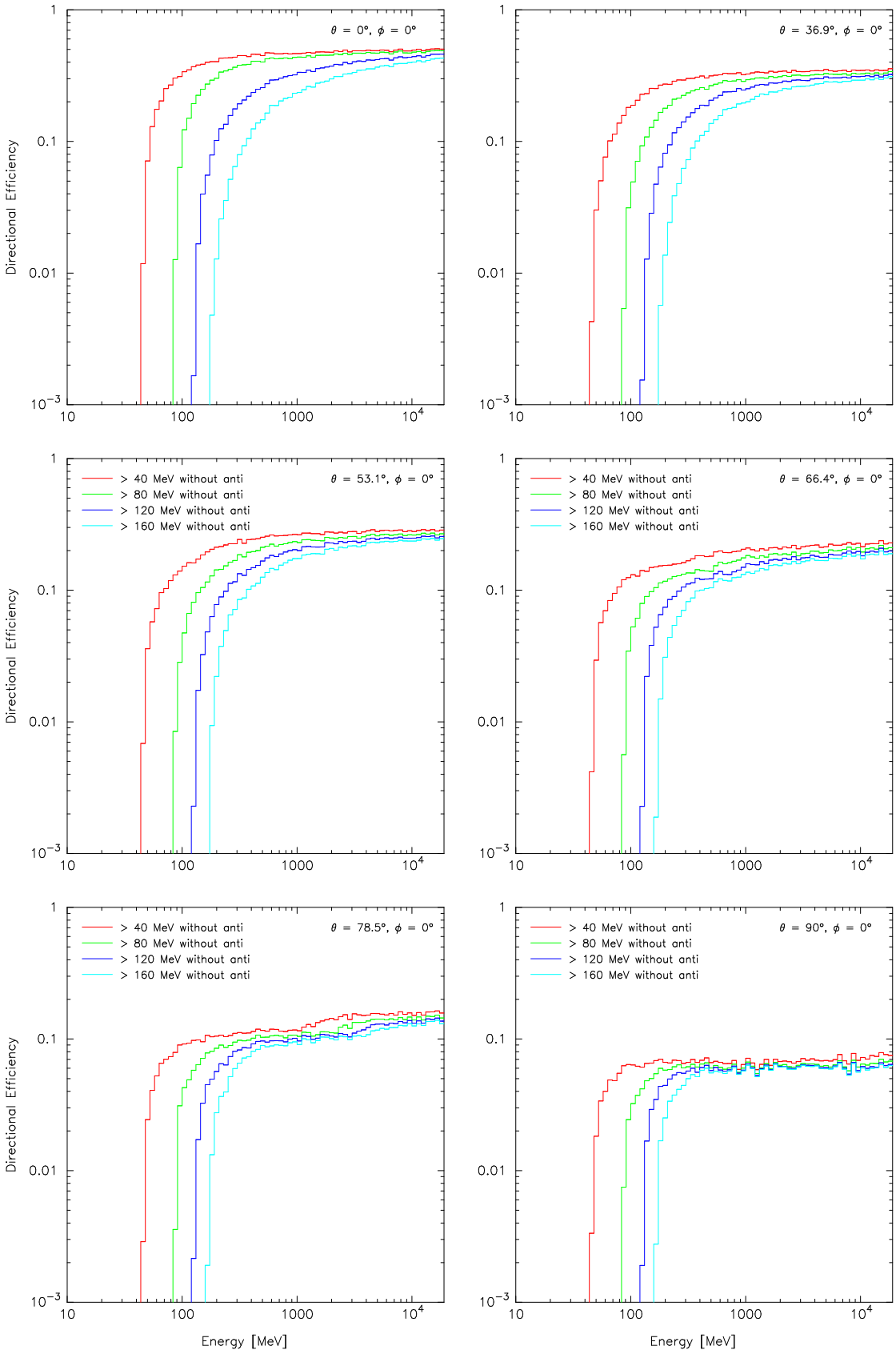


Figure 5.10: Simulated directional efficiencies for electrons. The plots show the directional efficiencies of the scintillator channels without anti.

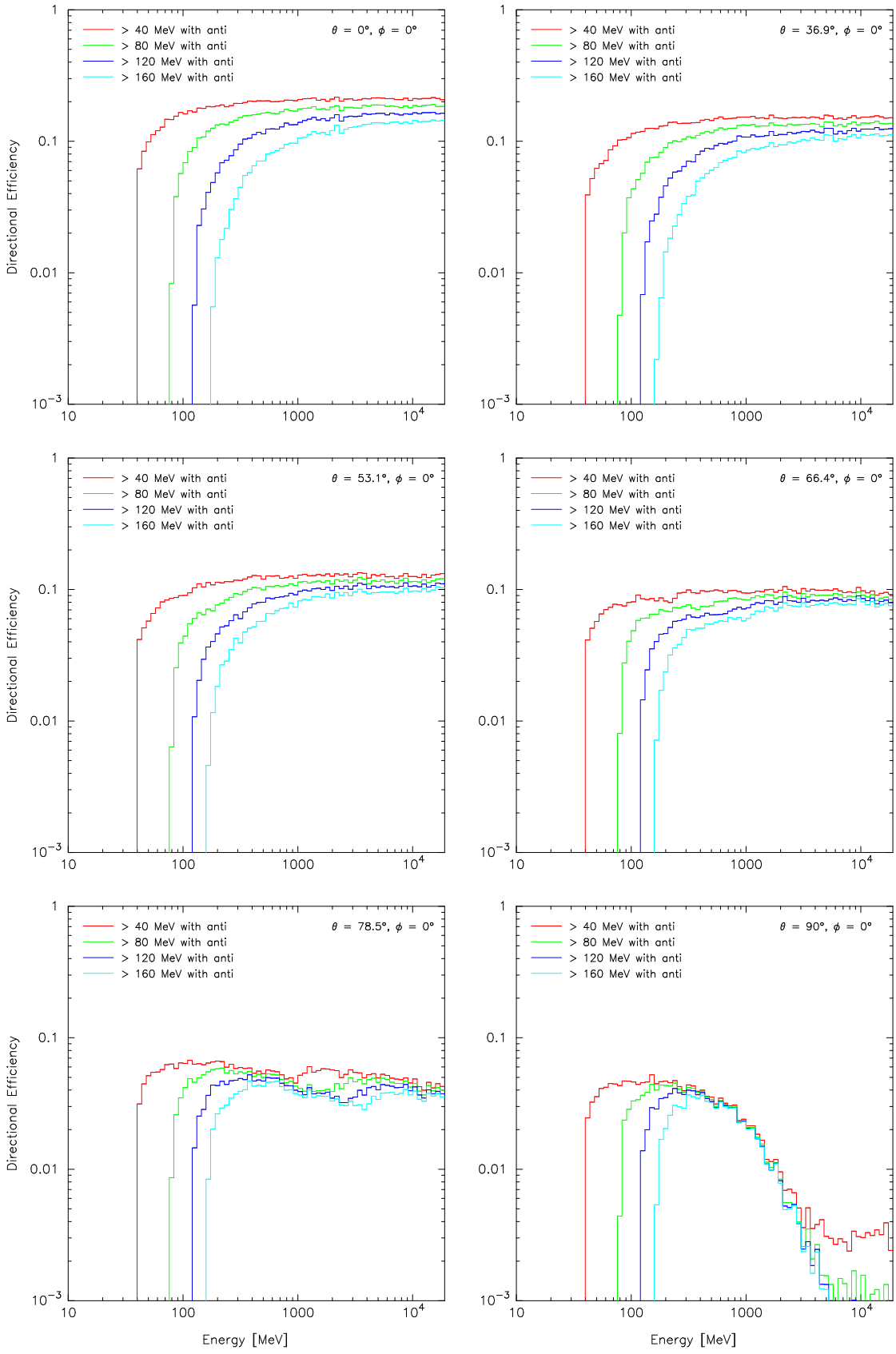


Figure 5.11: Simulated directional efficiencies for γ-rays. The plots show the directional efficiencies of the scintillator channels with anti.

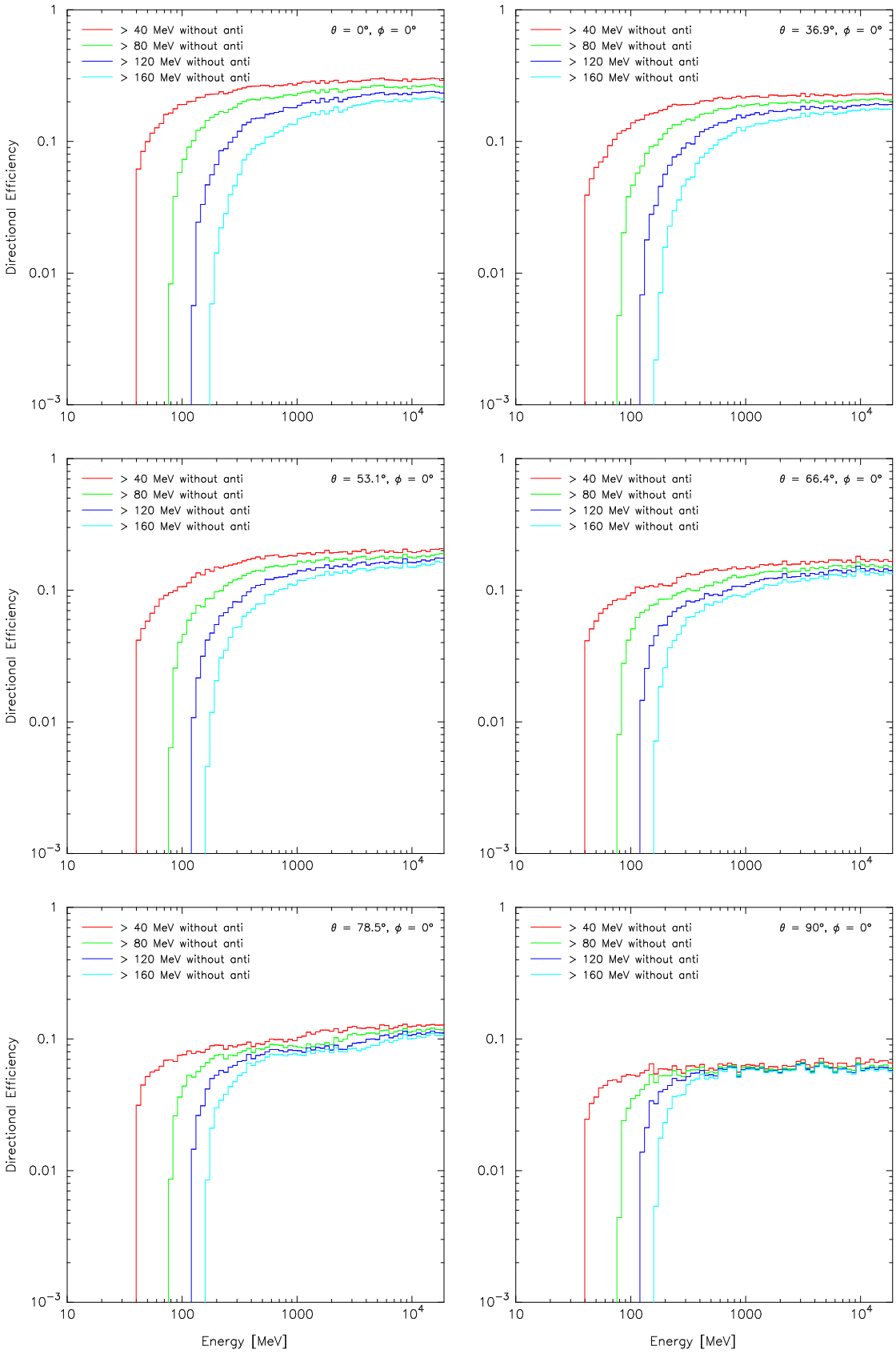


Figure 5.12: Simulated directional efficiencies for γ-rays. The plots show the directional efficiencies of the scintillator channels without anti.

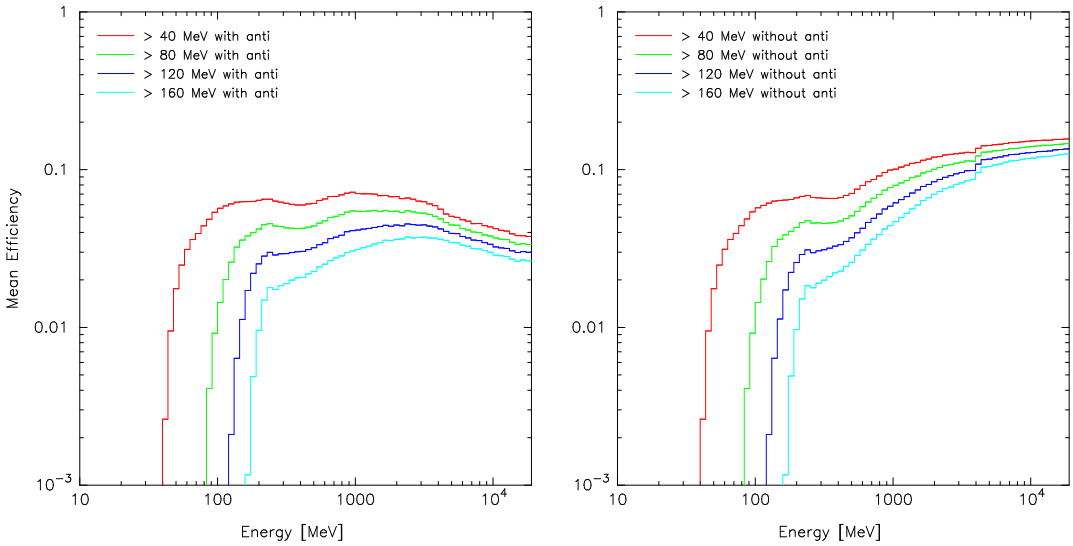


Figure 5.13: Simulated mean efficiencies of the scintillator channels for neutrons.

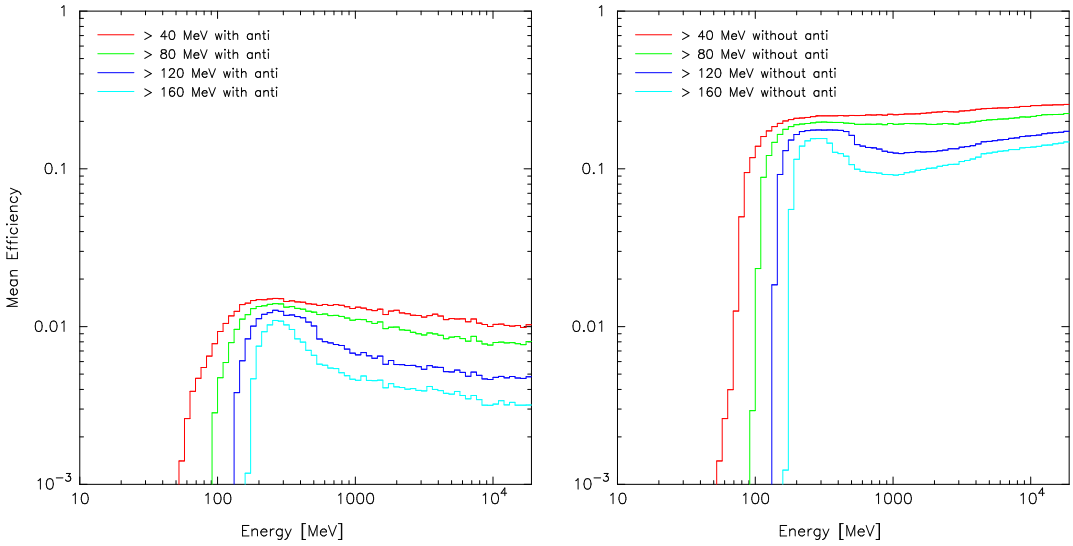


Figure 5.14: Simulated mean efficiencies of the scintillator channels for protons.

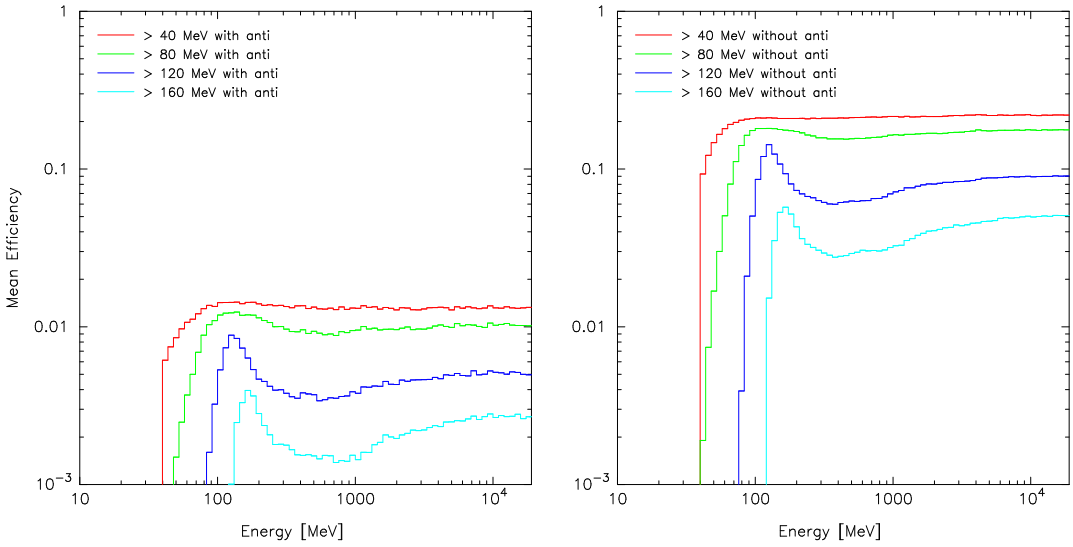


Figure 5.15: Simulated mean efficiencies of the scintillator channels for muons.

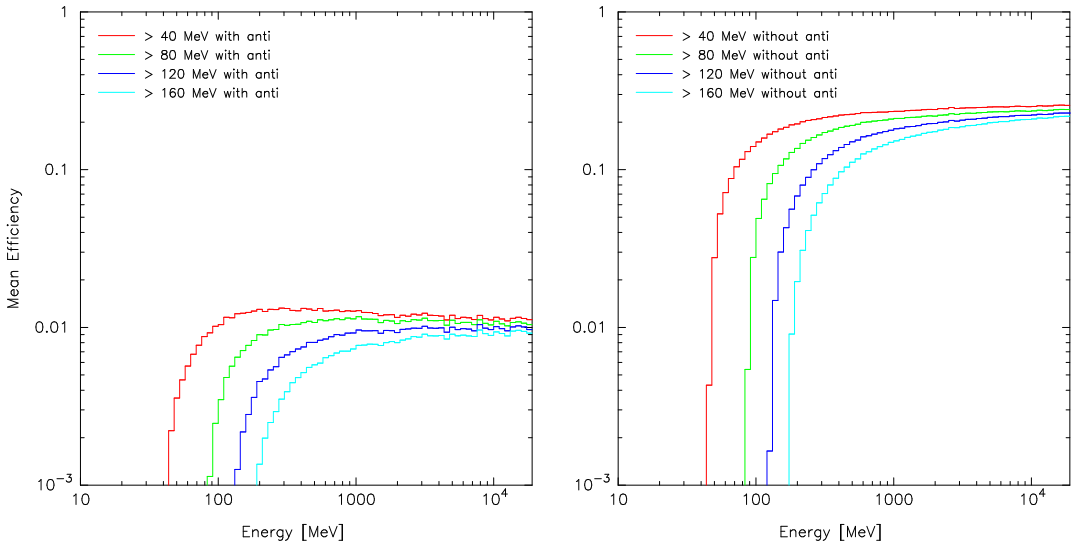


Figure 5.16: Simulated mean efficiencies of the scintillator channels for electrons.

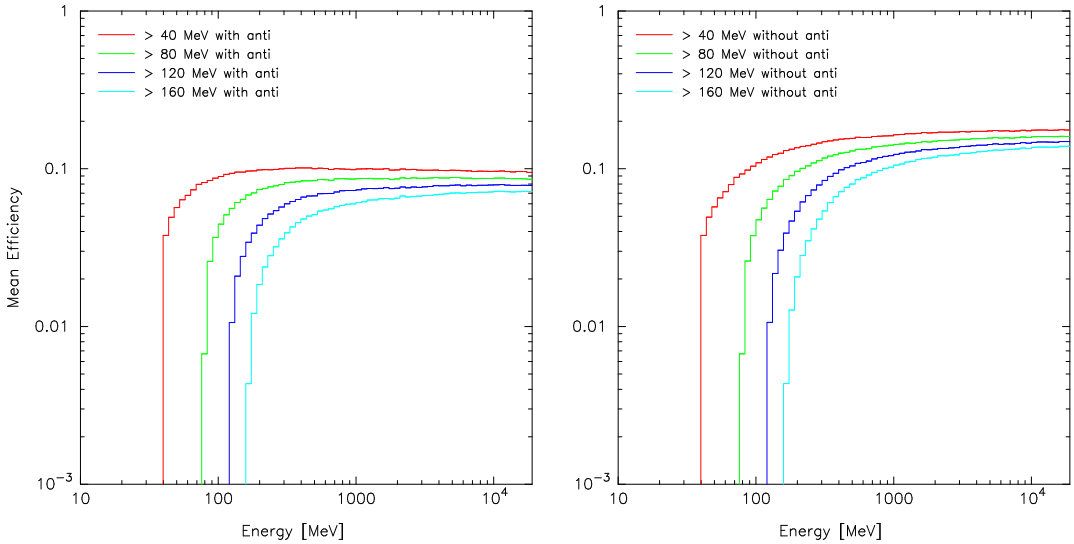


Figure 5.17: Simulated mean efficiencies of the scintillator channels for γ-rays.

5.1.2 Directional Counters

According to the electronic layout of the detector, it is possible that one incident particle can cause one count in multiple directional channels as described in Section 4.2.5. In order to investigate this effect, the efficiencies of the directional channels are subdivided according to the multiplicity m of the counts. Figure 5.18 shows the respective results for neutrons from the zenith direction.

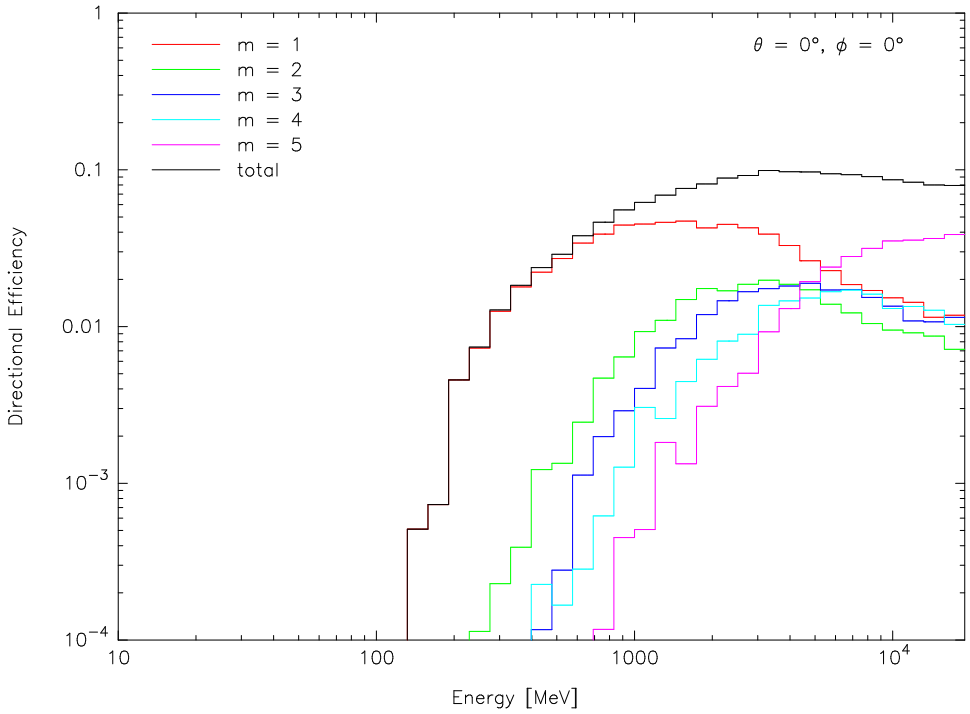


Figure 5.18: Simulated vertical directional efficiency of the directional counters for neutrons. The colored curves represent the contribution of the different multiplicities to the total efficiency (black).

Figures 5.19 to 5.23 show the directional efficiencies for five types of radiation in dependence of the zenith angle. Although the arrangement of the directional counters is obviously not rotationally symmetric, the dependence of the azimuth angle is small. Therefore, only the south direction is indicated.

The mean detection efficiencies of the directional counters are represented in Figures 5.24 to 5.28. These Figures show the total mean efficiency for the different radiation types.

In this work, we did not define a parameter to characterize the reliability of the directional information. It would be quite difficult to define such a parameter considering that due to the geometrical arrangement of the directional counters a unique determination of direction is impossible. Nevertheless, in Section 6.1.2 we will make a prediction of the resolution of the directional counters.

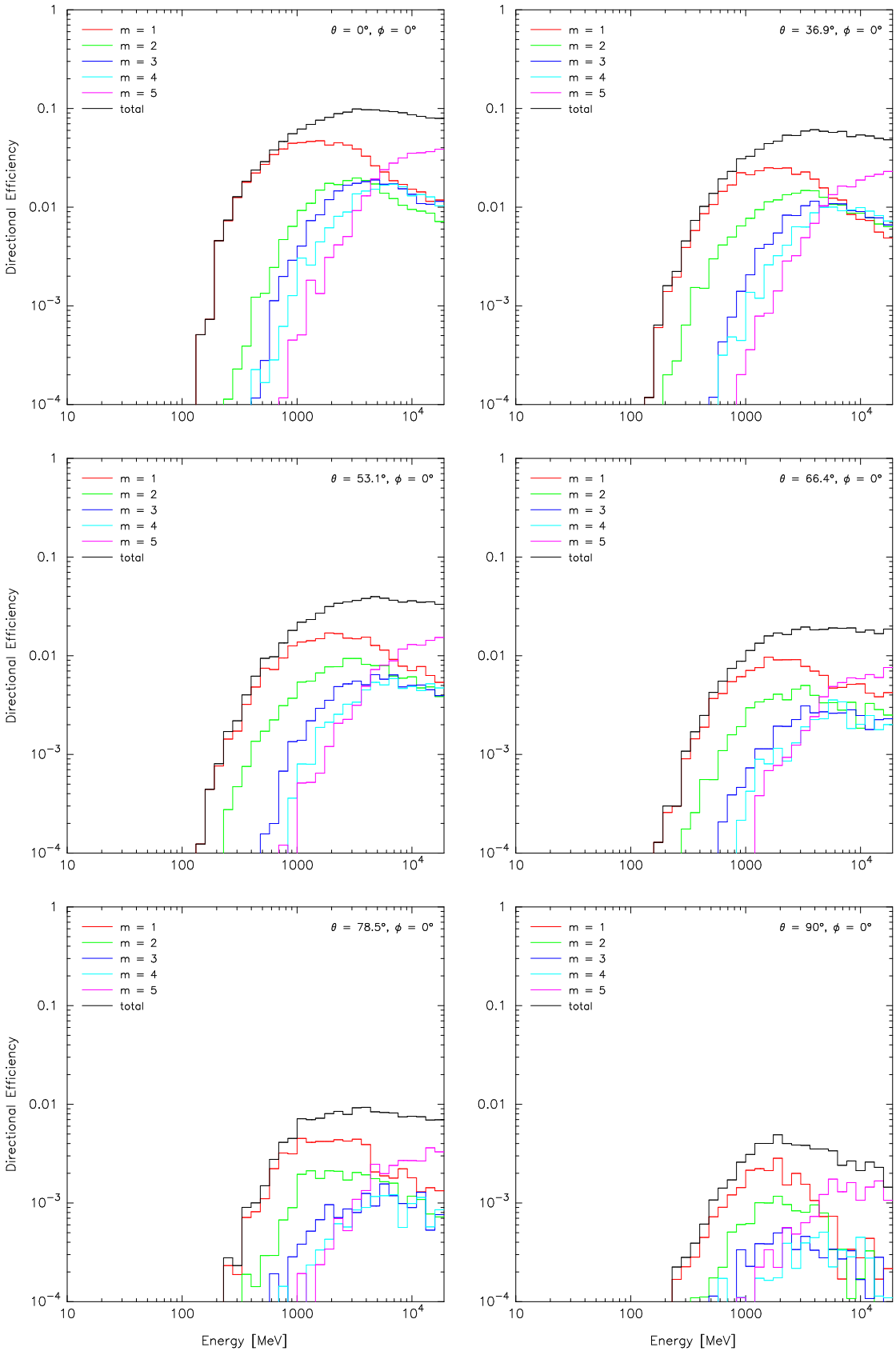


Figure 5.19: Simulated directional efficiencies for neutrons. The plots show the directional efficiencies of the direction determination channels.

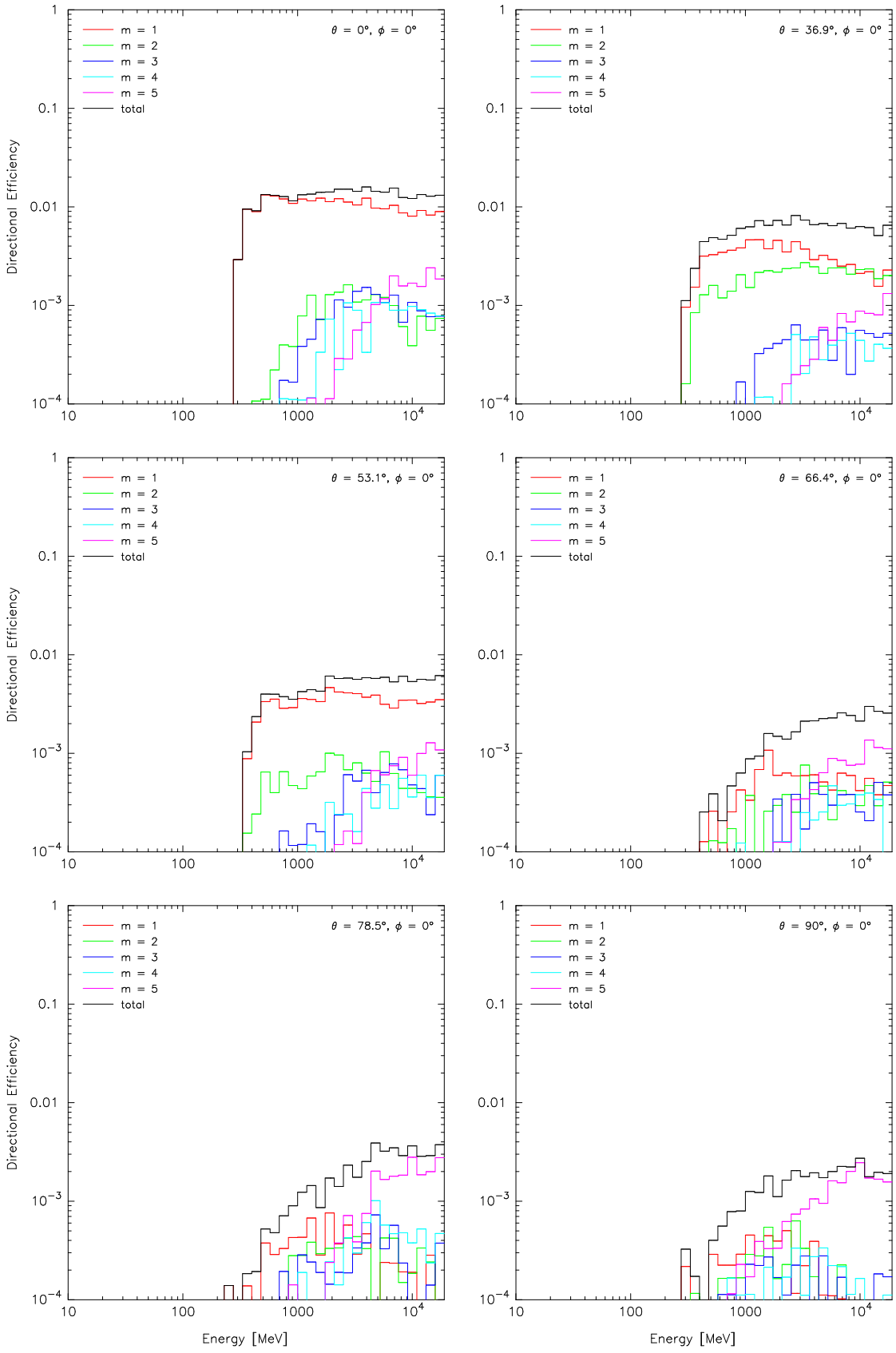


Figure 5.20: Simulated directional efficiencies for protons. The plots show the directional efficiencies of the direction determination channels.

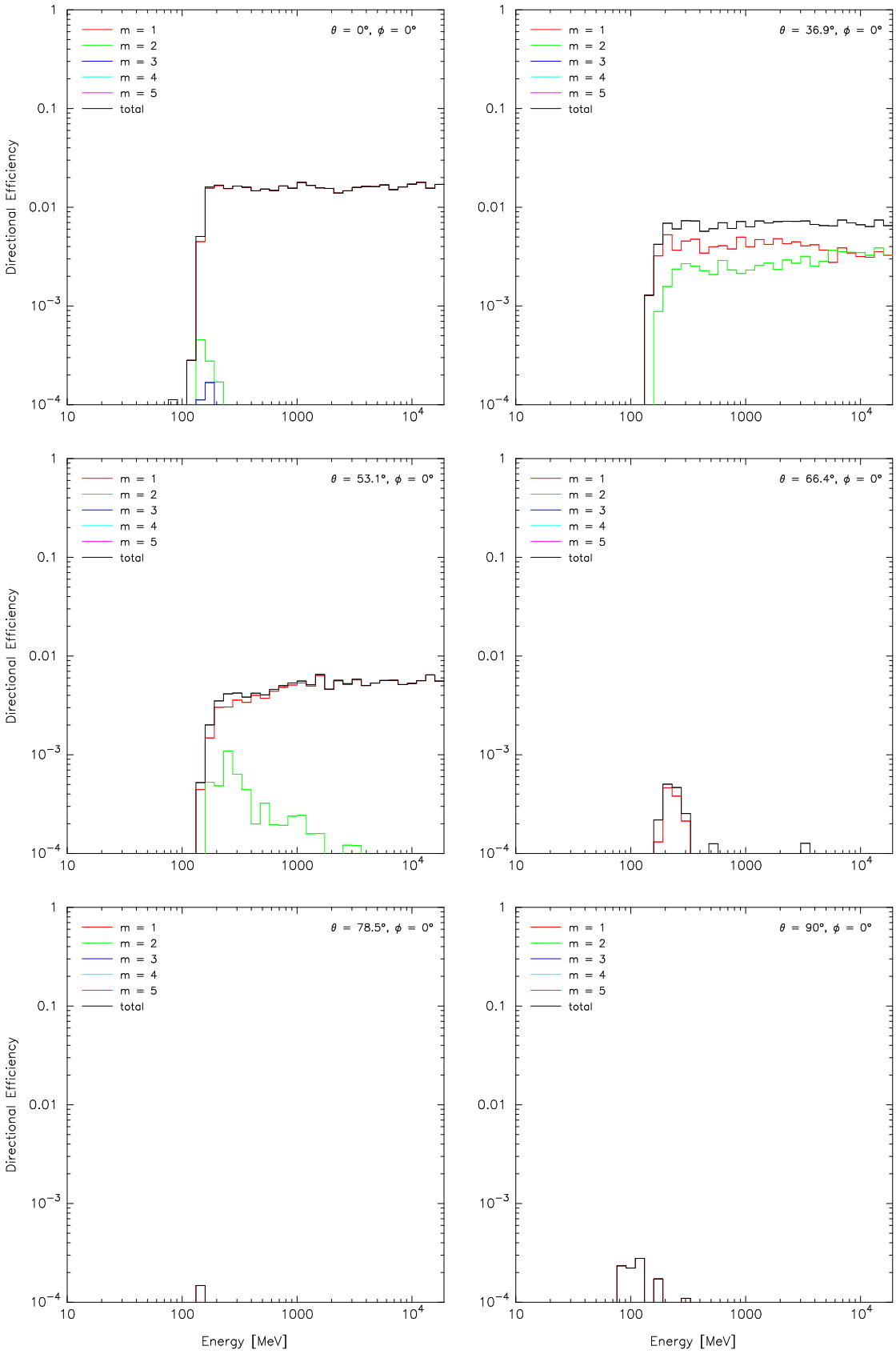


Figure 5.21: Simulated directional efficiencies for muons. The plots show the directional efficiencies of the direction determination channels.

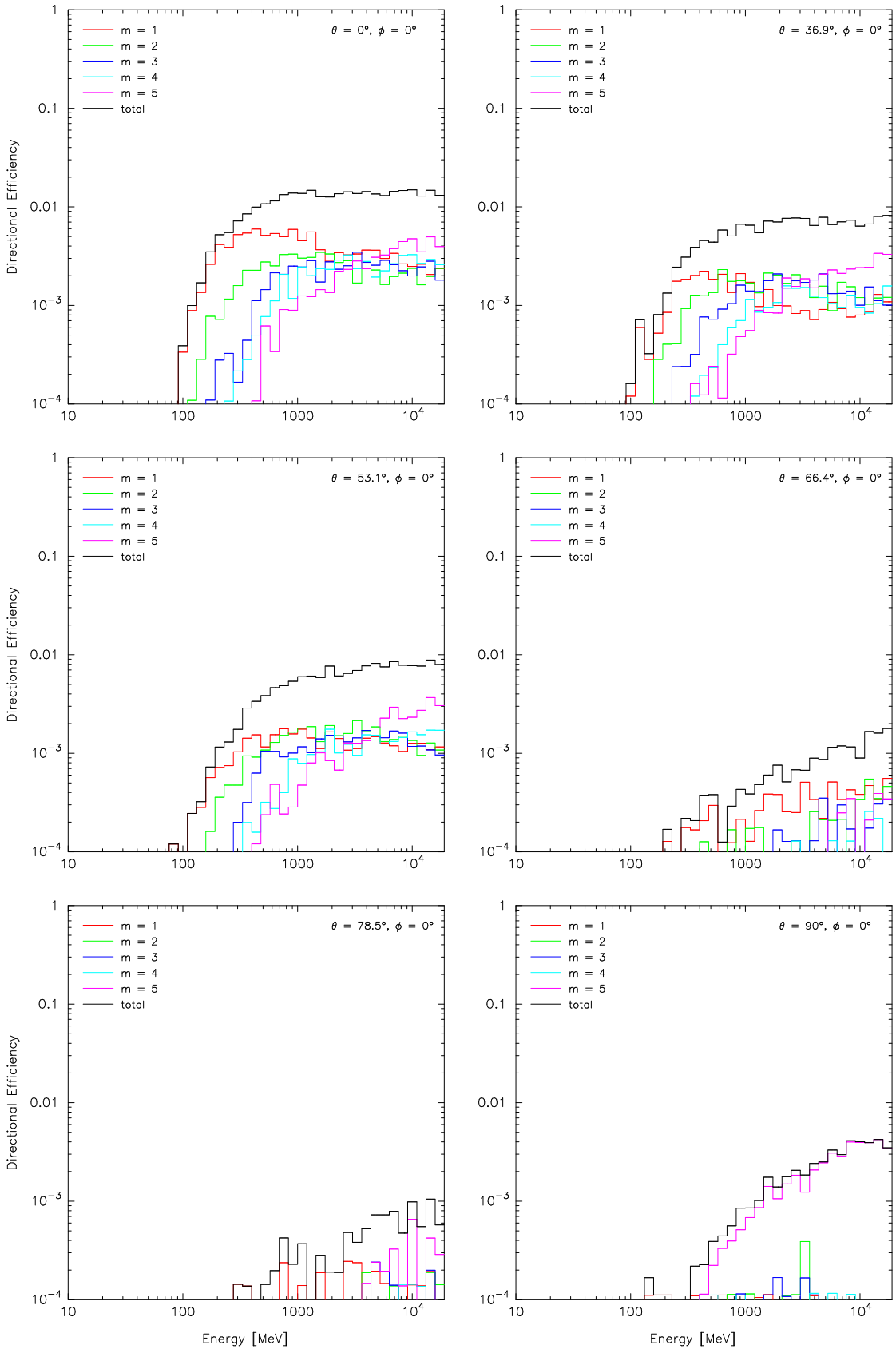


Figure 5.22: Simulated directional efficiencies for electrons. The plots show the directional efficiencies of the direction determination channels.

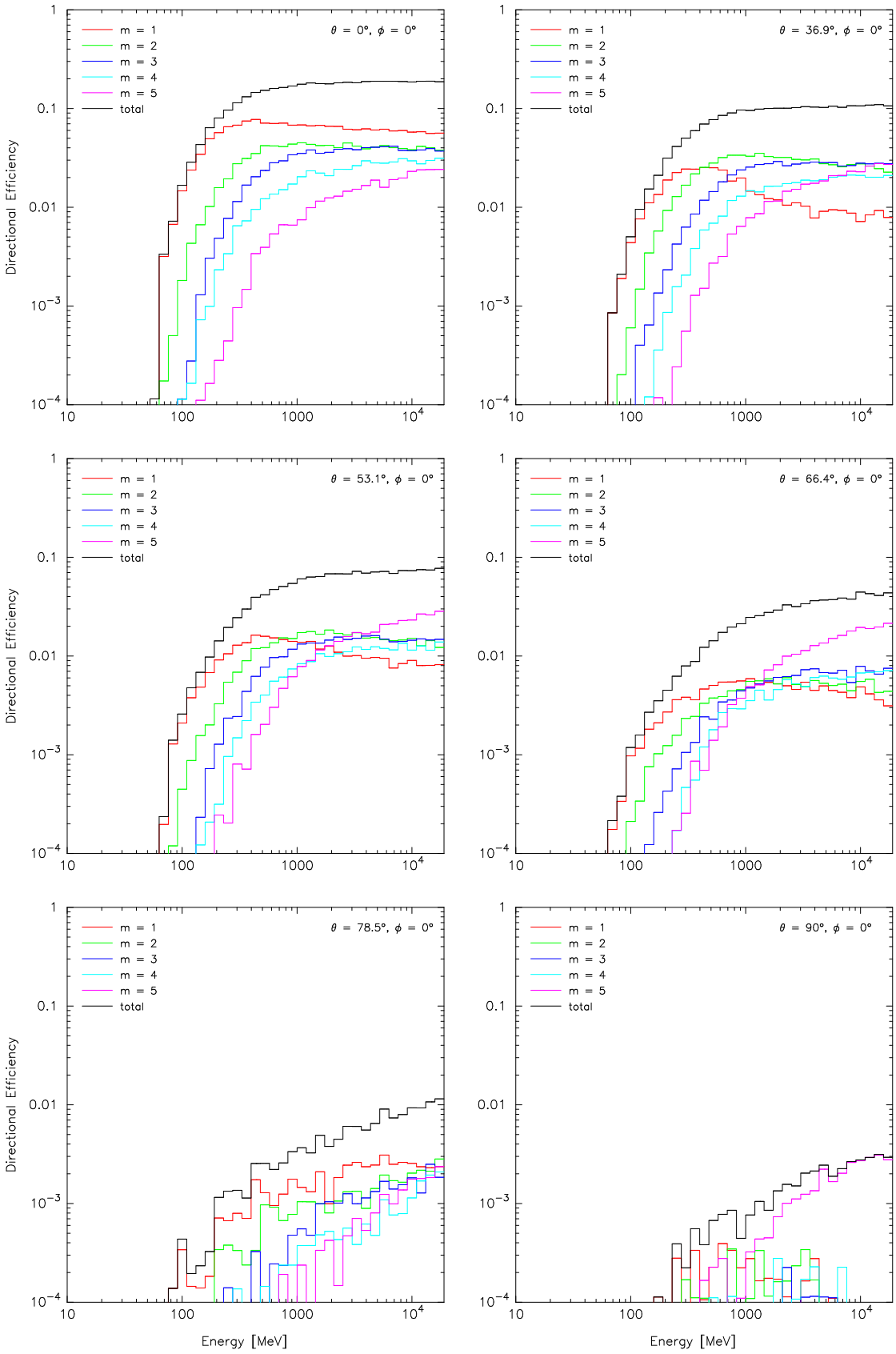


Figure 5.23: Simulated directional efficiencies for γ-rays. The plots show the directional efficiencies of the direction determination channels.

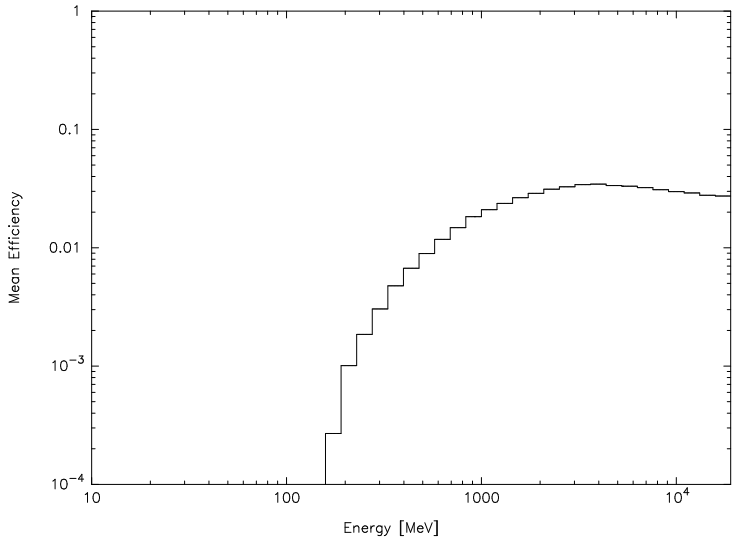


Figure 5.24: Simulated mean efficiency of the directional counters for neutrons.

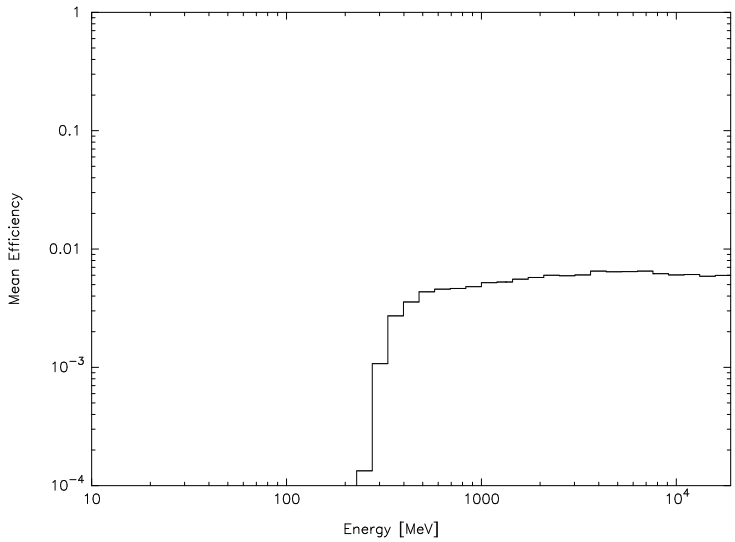


Figure 5.25: Simulated mean efficiency of the directional counters for protons.

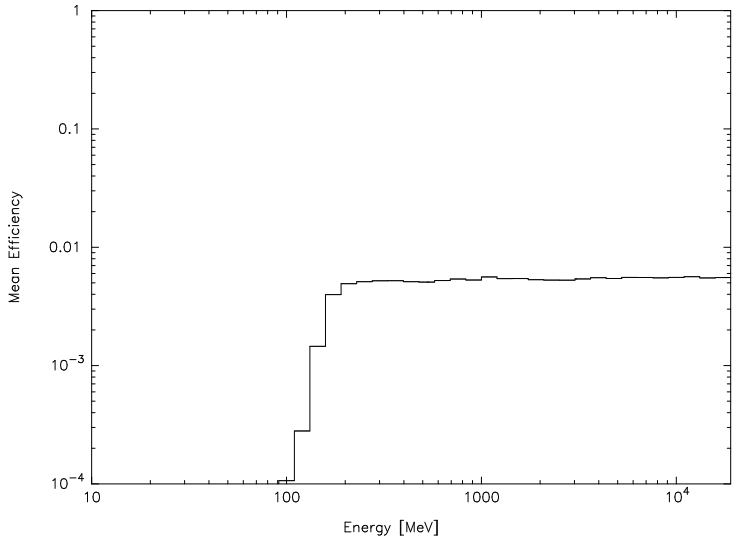


Figure 5.26: Simulated mean efficiency of the directional counters for muons.

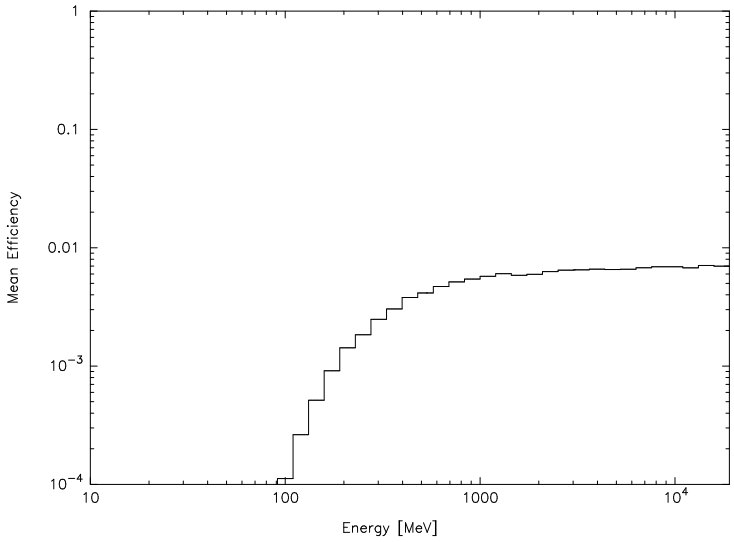


Figure 5.27: Simulated mean efficiency of the directional counters for electrons.

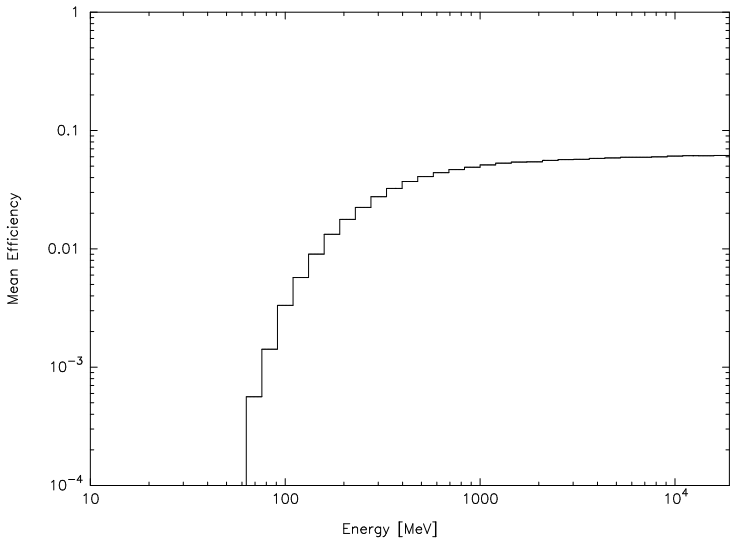


Figure 5.28: Simulated mean efficiency of the directional counters for γ-rays.

5.2 Counting Rate due to Galactic Cosmic Rays

In order to compare the simulation results with measurements the SONTEL counting rates N_i^n for galactic cosmic rays have been simulated. For this purpose the secondary energy spectra for mean solar activity, as introduced in Chapter 3, were used as the input. The simulation consisted of incident directions from six different zenith and seven azimuth angles, respectively. The counting rates per unit solid angle $\mathrm{d}N_i^n/\mathrm{d}\Omega$ of the detection channel n due to radiation type i, as obtained from the `SONTEL` GEANT3 application, are integrated numerically over the upper half space according to

$$N_i^n = \int\limits_{\cap} \left(\frac{\mathrm{d}N_i^n}{\mathrm{d}\Omega} \right)_{\theta,\phi} \mathrm{d}\Omega. \tag{5.3}$$

The total counting rate $\hat{N}^n$ of a detection channel n is obtained by summation over all radiation types

$$\hat{N}^n = \sum_i a_i \, N_i^n, \tag{5.4}$$

where the weighting factors a_i are first set equal to unity. Table 5.1 lists the contributions as well as the total counting rates of all detection channels of SONTEL.

Detection channel		N_n	N_p	N_μ	N_e	N_γ	$\hat{N}$
		[cts/10 s]					
$>$ 40 MeV		1216	46	913	65	763	**3003**
$>$ 80 MeV	with	679	37	637	31	313	**1697**
$>$ 120 MeV	anti	436	21	194	14	148	**813**
$>$ 160 MeV		292	13	58	7	80	**450**
$>$ 40 MeV		1452	695	13142	923	924	**17136**
$>$ 80 MeV	without	877	598	9836	482	396	**12189**
$>$ 120 MeV	anti	597	364	2822	239	199	**4220**
$>$ 160 MeV		423	250	836	132	113	**1754**
A = B − 2		58	2	51	2	26	**139**
A = B − 1		99	4	91	4	69	**267**
A = B	with anti	150	11	207	7	141	**516**
A = B + 1		99	4	91	4	69	**267**
A = B + 2		58	2	51	2	26	**139**
Veto counters		476	1190	27638	2826	827	**32956**

Table 5.1: Simulated galactic counting rates [cts/10 s] of all SONTEL detection channels. The table contains the partial counting rates of every considered radiation type as well as their sum. The results refer to mean solar activity.

5.3 Atmospheric Cascade of Solar Particles

The simulation of the atmospheric cascade has been executed with the `SOLSPEC` GEANT3 application. The obtained results were preprocessed for further use with application `SONTELSEP` for the simulation of the detector sensitivity to solar particles as will be discussed in Section 5.4.

Depending on the energy release and on the position of an eruption at the Sun, protons and neutrons can be injected in the interplanetary space, which possibly can have an effect on ground based cosmic ray detectors.

In order to simulate the SONTEL response to solar particles in a specific event, the secondary cosmic ray spectra produced by a given solar particle flux has to be known. The next two paragraphs treat the simulation of these secondary spectra for an altitude of 3135 m a.s.l. for the ground level enhancements of June 3, 1982, (neutron event) and April 15, 2001, (proton event), respectively. Both events are well documented in the literature (Chupp et al., 1987; Debrunner et al., 1983; Lockwood et al., 2002; Chilingarian, 2002). In addition the April 15, 2001, event was actually observed at Gornergrat.

5.3.1 Solar Neutron Event of June 3, 1982

During a period of high solar activity, a class X8 white light flare with associated emissions from meter radio wave to γ-rays erupted in McMath Region 3763 (S09E72) on June 3, 1982, at 11:43 UT. Neutrons produced within this flare were the first solar neutrons detected at Earth as identified by neutron monitor measurements at Jungfraujoch, Lomnický Štít, and Rome (Chupp et al., 1987).

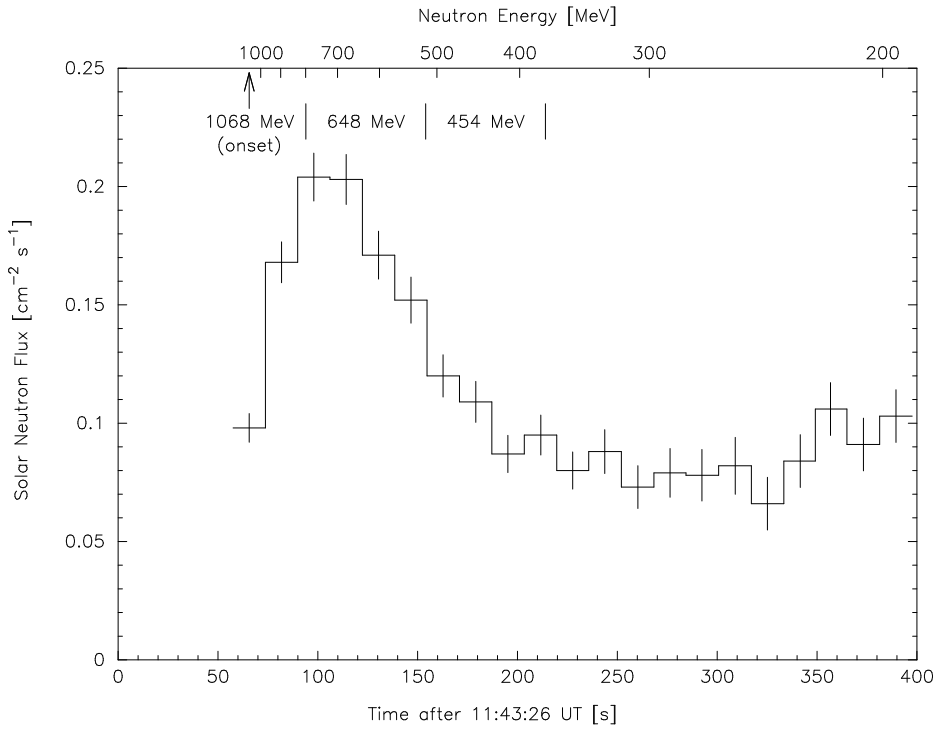


Figure 5.29: Primary solar neutron flux J_n at the top of the atmosphere after the June 3, 1982, solar neutron event. Neutrons with lower kinetic energies arrived with increasing time delay.

The primary neutron fluxes J_n at the top of the atmosphere that were used as input for our simulations were those evaluated by Flückiger (private communication, 2002) based on neutron monitor data from Jungfraujoch. Assuming that the neutrons were emitted at the Sun as a δ-function in time, the time delay Δt between the observation of the impulsive γ-ray emission and the arrival of the neutrons at Earth depends on the kinetic energy T_n of the neutrons

$$\Delta t = \frac{1\,\mathrm{AU}}{c}\left(\frac{1}{\beta} - 1\right) \text{ with } \beta = \sqrt{1 - \frac{1}{\gamma^2}}, \tag{5.5}$$

where the Lorentz factor γ is defined as

$$\gamma = 1 + \frac{T_\mathrm{n}}{m_\mathrm{n} c^2}\,. \tag{5.6}$$

The time delay Δt is then used as an indicator of the kinetic energy T_n of the neutrons. Considering the detector sensitivity S_n of the neutron monitor (Stein, 1989) the primary neutron fluxes J_n can then be determined from the increase in the neutron monitor data. Figure 5.29 shows the calculated neutron fluxes as a function of time and energy. The plot shows only the statistical uncertainties, while the values are estimated to be reliable within a factor of 2.

The atmospheric cascade of primary solar neutrons was calculated for three different energies: the onset energy (1068 MeV), the mean energies corresponding to the time intervals 11:45:00–11:46:00 UT (648 MeV) and 11:46:00–11:47:00 UT (454 MeV). Because of the onset time around local noon the zenith angle of the Sun was almost minimal $\theta = 25.5°$. The azimuth angle at this time was $\phi = 167°$W. Table 5.2 contains the simulated integral flux of the secondary particles at Gornergrat altitude for the three primary energies.

Radiation type	$J_\mathrm{n}(1068\,\mathrm{MeV})$ $[\mathrm{cm^{-2}\,s^{-1}}]$	$J_\mathrm{n}(648\,\mathrm{MeV})$ $[\mathrm{cm^{-2}\,s^{-1}}]$	$J_\mathrm{n}(454\,\mathrm{MeV})$ $[\mathrm{cm^{-2}\,s^{-1}}]$
Primary Neutrons	9.8×10^{-2}	1.83×10^{-1}	1.13×10^{-1}
Neutrons	1.32×10^{-2}	1.34×10^{-2}	5.88×10^{-3}
Protons	9.94×10^{-4}	6.71×10^{-4}	2.12×10^{-4}
γ-rays	5.65×10^{-3}	4.24×10^{-3}	1.72×10^{-3}
Electrons & Positrons	4.12×10^{-4}	1.60×10^{-4}	4.14×10^{-5}
Muons	4.87×10^{-5}	1.18×10^{-5}	1.35×10^{-7}
Pions	—	—	—
Neutrinos	8.81×10^{-2}	4.24×10^{-3}	1.72×10^{-3}

Table 5.2: Integral primary neutron flux $J_\mathrm{n}(T_\mathrm{n})$ at the top of the atmosphere and simulated secondary particle fluxes at an altitude of 3135 m a.s.l. for the June 3, 1982, solar neutron event.

Figure 5.30 shows the omnidirectional differential energy spectra for the most abundant particle types of the atmospheric cascade.

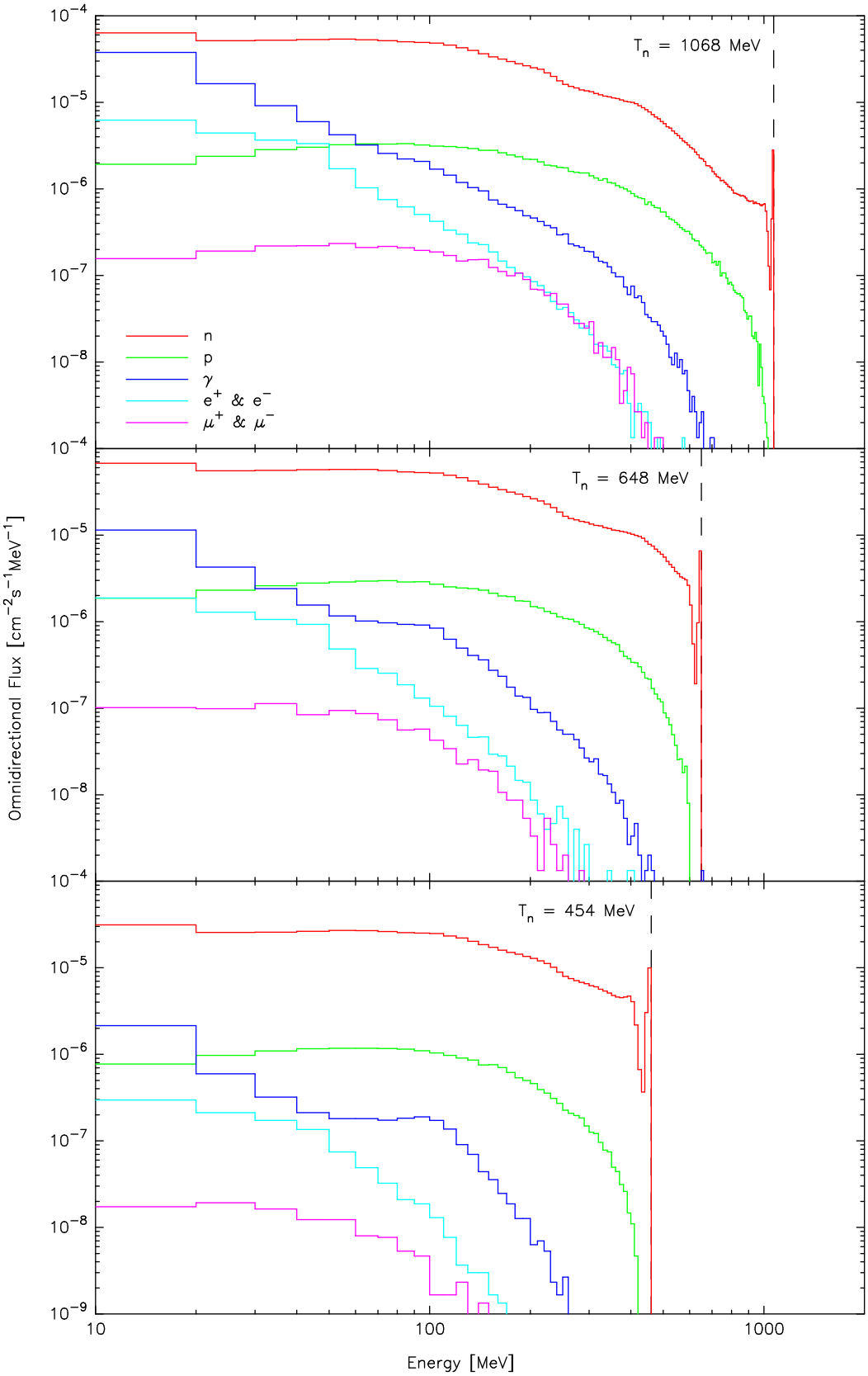


Figure 5.30: Evolution of the omnidirectional spectra of secondary particles at an altitude of 3135 m a.s.l. for three different primary neutron energies (1068, 648, and 454 MeV) as calculated for the June 3, 1982, event.

5.3.2 Solar Proton Event of April 15, 2001

Lockwood et al. (2002) determined the proton spectrum at the top of the atmosphere for this and further events. They assumed a power law spectrum of the primary solar protons and determined the spectral parameters from neutron monitor data. For 14:17 UT on April 15, 2001, they obtained a differential rigidity spectrum described by

$$\frac{\mathrm{d}J_\mathrm{p}}{\mathrm{d}P\,\mathrm{d}\Omega} = 4.9 \times 10^5\, P^{-6.4}\ [\mathrm{m^{-2}\,s^{-1}\,sr^{-1}\,GV^{-1}}].$$

For our purpose this solar proton rigidity spectrum was transformed into an energy spectrum and was implemented as a table in the `SOLSPEC` GEANT3 application.

For the simulation of the secondary particle fluxes the differential solar proton spectrum has to be integrated over the rigidity range covered by the application. As the lower limit, the *effective vertical cutoff rigidity* for Gornergrat and for the time of the event, $P_\mathrm{c} = 4.64\,\mathrm{GV}$, is used (Bütikofer, private communication, 2002). The upper boundary is set to $P_\mathrm{max} = 20\,\mathrm{GV} \sim 20\,\mathrm{GeV}$ as for all the simulations in this work. Furthermore, it is assumed that all protons hit the Earth's atmosphere below a zenith angle smaller than $\theta_\mathrm{max} = 30°$. Thus, the integrated solar proton flux is estimated by

$$J_\mathrm{p} = 2\pi \int_0^{\theta_\mathrm{max}} \int_{P_\mathrm{c}}^{P_\mathrm{max}} \frac{\mathrm{d}J_\mathrm{p}}{\mathrm{d}P\,\mathrm{d}\Omega}\, \cos\theta\,\mathrm{d}P\, \sin\theta\,\mathrm{d}\theta = 1.8 \times 10^{-3}\,\mathrm{cm^{-2}\,s^{-1}}.$$

Figure 5.31 shows the resulting omnidirectional secondary flux spectra of the most abundant species of secondary particles at 3135 m a.s.l. assuming the input parameters given above.

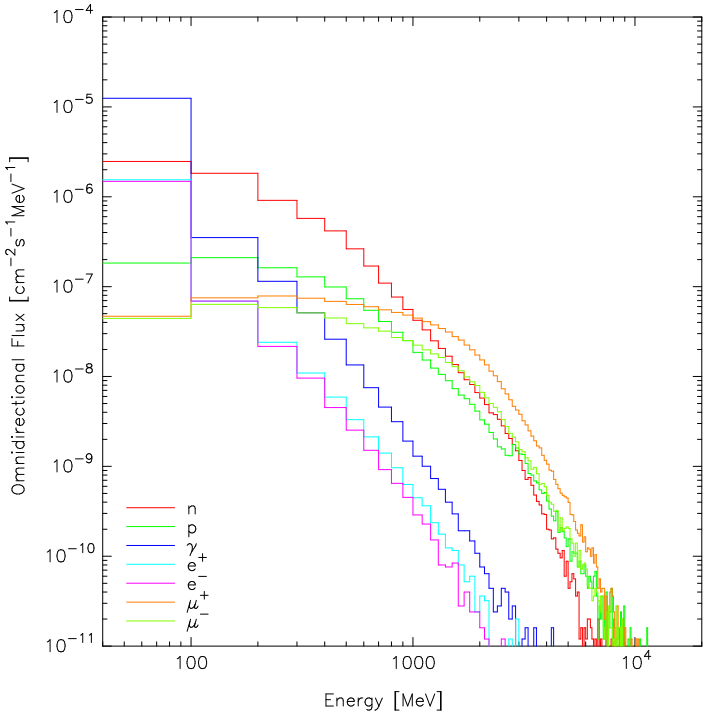


Figure 5.31: Simulated omnidirectional spectra of the most abundant secondary particles at an altitude of 3135 m a.s.l. for the April 15, 2001, GLE (14:17 UT).

5.4 Detector Response for Solar Particles

Using the particle spectra presented in the previous chapter as input, the increases of the counting rates during the June 3, 1982, and April 15, 2001, solar particle events were simulated with the `SONTELSEP` GEANT3 application. The results are compiled in the next two paragraphs.

5.4.1 Solar Neutron Event of June 3, 1982

Table 5.3 lists the simulated increase in the different counting rates after the June 3, 1982, solar neutron event. The table shows the contributions ΔN_i^n of the particle types i to the total increase ΔN^n of the detection channel n, defined in analogy to (5.4) as

$$\Delta N^n = \sum_i a_i \, \Delta N_i^n, \tag{5.7}$$

where the weighting factors a_i are also set equal to unity. The second last column of the table shows the relative increase $\Delta N/\hat{N}$, expressed in percentage of the galactic counting rate $\hat{N}$. The significance of the increase $\Delta N/\sigma(\hat{N})$, where the standard deviation $\sigma(\hat{N})$ of the galactic counting rate is given by

$$\sigma(\hat{N}) = \frac{1}{\sqrt{\hat{N}[\mathrm{cts}/10\,\mathrm{s}]}}, \tag{5.8}$$

is summarized in the last column of the table.

	Detection channel		ΔN_n	ΔN_p	ΔN_μ	ΔN_e	ΔN_γ	ΔN	$\frac{\Delta N}{\hat{N}}$	$\frac{\Delta N}{\sigma(\hat{N})}$
$T_\mathrm{n} = \mathbf{1068\,MeV}$	> 40 MeV	with anti	259	2	0	1	23	**285**	9.5 %	5.2
	> 80 MeV		100	1	0	0	11	**112**	6.6 %	2.7
	> 120 MeV		42	1	0	0	6	**49**	6.0 %	1.7
	> 160 MeV		18	1	0	0	3	**22**	4.9 %	1.0
	> 40 MeV	without anti	266	26	2	14	30	**338**	2.0 %	2.6
	> 80 MeV		104	19	2	10	15	**150**	1.2 %	1.4
	> 120 MeV		45	13	1	8	8	**75**	1.8 %	1.2
	> 160 MeV		20	8	0	6	5	**39**	2.2 %	0.9
	$\mathrm{A} = \mathrm{B} - 2$	with anti	0	0	0	0	0	**0**	0.0 %	0.0
	$\mathrm{A} = \mathrm{B} - 1$		1	0	0	0	1	**2**	0.7 %	0.1
	$\mathrm{A} = \mathrm{B}$		3	0	0	0	1	**4**	0.8 %	0.2
	$\mathrm{A} = \mathrm{B} + 1$		1	0	0	0	1	**2**	0.7 %	0.1
	$\mathrm{A} = \mathrm{B} + 2$		0	0	0	0	0	**0**	0.0 %	0.0
	Veto counters		58	92	7	69	53	**279**	0.9 %	1.5
$T_\mathrm{n} = \mathbf{648\,MeV}$	> 40 MeV	with anti	226	2	0	0	5	**233**	7.8 %	4.3
	> 80 MeV		92	1	0	0	2	**95**	5.6 %	2.3
	> 120 MeV		41	1	0	0	1	**43**	5.3 %	1.5
	> 160 MeV		18	0	0	0	1	**19**	4.2 %	0.9
	> 40 MeV	without anti	233	24	0	3	6	**266**	1.6 %	2.0
	> 80 MeV		96	17	0	2	3	**118**	1.0 %	1.1
	> 120 MeV		43	12	0	1	1	**57**	1.4 %	0.9
	> 160 MeV		20	7	0	1	1	**29**	1.7 %	0.7
	$\mathrm{A} = \mathrm{B} - 2$	with anti	0	0	0	0	0	**0**	0.0 %	0.0
	$\mathrm{A} = \mathrm{B} - 1$		1	0	0	0	0	**1**	0.4 %	0.1
	$\mathrm{A} = \mathrm{B}$		3	0	0	0	0	**3**	0.6 %	0.1
	$\mathrm{A} = \mathrm{B} + 1$		1	0	0	0	0	**1**	0.4 %	0.1
	$\mathrm{A} = \mathrm{B} + 2$		0	0	0	0	0	**0**	0.0 %	0.0
	Veto counters		50	84	2	12	11	**159**	0.5 %	0.9
$T_\mathrm{n} = \mathbf{454\,MeV}$	> 40 MeV	with anti	96	0	0	0	1	**97**	3.2 %	1.8
	> 80 MeV		37	0	0	0	0	**37**	2.2 %	0.9
	> 120 MeV		15	0	0	0	0	**15**	1.8 %	0.5
	> 160 MeV		6	0	0	0	0	**6**	1.3 %	0.3
	> 40 MeV	without anti	98	7	0	0	1	**106**	0.6 %	0.8
	> 80 MeV		38	5	0	0	0	**43**	0.4 %	0.4
	> 120 MeV		16	3	0	0	0	**19**	0.5 %	0.3
	> 160 MeV		6	2	0	0	0	**8**	0.5 %	0.2
	$\mathrm{A} = \mathrm{B} - 2$	with anti	0	0	0	0	0	**0**	0.0 %	0.0
	$\mathrm{A} = \mathrm{B} - 1$		0	0	0	0	0	**0**	0.0 %	0.0
	$\mathrm{A} = \mathrm{B}$		1	0	0	0	0	**1**	0.2 %	0.0
	$\mathrm{A} = \mathrm{B} + 1$		0	0	0	0	0	**0**	0.0 %	0.0
	$\mathrm{A} = \mathrm{B} + 2$		0	0	0	0	0	**0**	0.0 %	0.0
	Veto counters		20	28	0	0	2	**50**	0.2 %	0.3

Table 5.3: Simulated increase in the various counting rates [cts/10 s] caused by solar neutrons during the June 3, 1982, event. The tables list the results for three different primary neutron energies and fluxes as discussed in Section 5.3.1.

5.4.2 Solar Proton Event of April 15, 2001

The simulated increase in the counting rates during the solar proton event of April 15, 2001, around 14:17 UT is shown in Table 5.4. The table shows the contributions of the different kinds of radiation, the total increase according to (5.7), the relative increase $\Delta N/\hat{N}$ of the counting rate, and also the significance $\Delta N/\sigma(\hat{N})$ of the 10-second counting rates.

Detection channel		ΔN_{n}	ΔN_{p}	ΔN_{μ}	ΔN_{e}	ΔN_{γ}	ΔN	$\frac{\Delta N}{\hat{N}}$	$\frac{\Delta N}{\sigma(\hat{N})}$
$>$ 40 MeV		35	0	3	0	7	**45**	1.5 %	0.8
$>$ 80 MeV	with	18	0	2	0	4	**24**	1.4 %	0.6
$>$ 120 MeV	anti	10	0	0	0	2	**12**	1.5 %	0.4
$>$ 160 MeV		6	0	0	0	1	**7**	1.6 %	0.3
$>$ 40 MeV		37	1	41	5	9	**93**	0.5 %	0.7
$>$ 80 MeV	without	20	1	29	3	5	**58**	0.5 %	0.5
$>$ 120 MeV	anti	12	1	2	1	2	**18**	0.4 %	0.3
$>$ 160 MeV		7	0	0	1	1	**9**	0.5 %	0.2
A = B − 2		0	0	0	0	0	**0**	0.0 %	0.0
A = B − 1	with	1	0	0	0	1	**2**	0.7 %	0.1
A = B	anti	2	0	1	1	0	**4**	0.8 %	0.2
A = B + 1		1	0	0	0	1	**2**	0.7 %	0.1
A = B + 2		0	0	0	0	0	**0**	0.0 %	0.0
Veto counters		9	4	58	10	8	**89**	0.3 %	0.5

Table 5.4: Simulated increase in the various counting rates [cts/10 s] caused by solar protons around 14:17 UT during the April 15, 2001, event. The table shows the contributions made by the different species of particles to the total counting rate increase.

Chapter 6

Discussion and Conclusion

In this chapter we discuss the SONTEL simulation results obtained from our GEANT3 applications as presented in the previous chapter. This discussion includes a comparison with outcomes obtained by other authors and with SONTEL data.

6.1 Detection Efficiencies

6.1.1 Scintillators

Most of the directional (5.1) and mean detection efficiencies (5.2) of SONTEL (Figures 5.3 to 5.17) show a similar structure in the energy dependence. The efficiency increases rapidly for radiation with energy above the discrimination level, and in general shows only a small energy dependence above $\sim 1\,\mathrm{GeV}$. However, characteristic deviations of this general behavior can be identified for different types of radiation. In the following we analyze the detection efficiencies of the scintillators for the different types of radiation.

Neutrons

Both the detection channels with (Figure 5.3) and without anti (Figure 5.4) have a quite similar efficiency from the lowest energies up to $\sim 500\,\mathrm{MeV}$. While above $\sim 500\,\mathrm{MeV}$ the efficiency of the channels without anti is still increasing with increasing neutron energy, the efficiency of the channels with anti decreases, in particular for directions of incidence with large zenith angles. This effect is due to the fact that in the scintillator material, neutrons with energies above $\sim 500\,\mathrm{MeV}$ produce charged secondary particles which can penetrate the veto counters and cause a veto signal *after* the signal from the photomultipliers. The simulations show that the scattering in the direction of the secondary particle increases with increasing neutron energy. In addition, the probability of a secondary particle to hit the veto counters is greater for larger zenith angles of the incident neutrons due to the geometry of the detector.

From the comparison of the results summarized in Figures 5.3 and 5.4 it can be concluded that the identification of neutrons by the veto system is optimum only for energies below $\sim 500\,\mathrm{MeV}$ and zenith angles $\lesssim 50°$. At higher energies an increasing fraction of neutrons is identified as charged particles. At 2 GeV this fraction is of the order of $\sim 50\%$.

Protons

The comparison of Figures 5.5 and 5.6 shows that over the entire energy range the efficiency of the detection channels with anti is lower by more than one order of magnitude than the efficiency of the channels without anti counters.

It is remarkable that for protons with small zenith angles the efficiency of the energy channels $> 120\,\mathrm{MeV}$ and $> 160\,\mathrm{MeV}$ show a step-like decrease at $\sim 300\,\mathrm{MeV}$. This decrease is a consequence of the particle interactions in the detector and the specific settings of the discriminator levels. Figure 6.1 represents the distribution of the deposited energy within the scintillators by protons from the zenith direction. Due to the energy dependence of the stopping power $\mathrm{d}E/\mathrm{d}x$, the deposited energy within the scintillator by protons decreases with increasing particle energy above 300 MeV, i.e. the peaks in

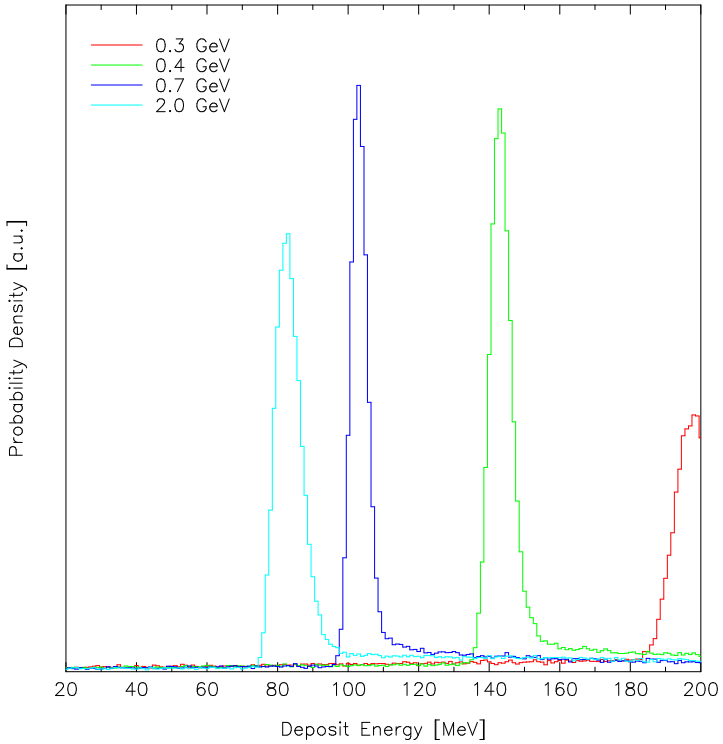


Figure 6.1: Distribution of deposited energy within the scintillator by protons from the zenith direction. The different curves refer to different initial particle energies as indicated.

Figure 6.1 are displaced to lower energies. For a wide range of initial kinetic energies the mean energy deposited is between the threshold energies 80 MeV and 120 MeV. Protons with initial kinetic energy above $\sim 600\,\mathrm{MeV}$ cannot be detected by the two higher energy channels despite the fact that their initial energy is well above the respective threshold energies. With increasing zenith angles the traversed distance of the protons within the scintillator increases and therefore the total deposited energy as well. Consequently, the effect disappears for incident directions with larger zenith angles.

Muons

Again, the comparison of the directional efficiencies of the energy channels without and with anti (Figures 5.7 and 5.8) shows a difference by one order of magnitude or more.

The muon efficiencies also show the effect described above for protons, i.e. the step-like decrease in the energy channels $> 120\,\mathrm{MeV}$ and $> 160\,\mathrm{MeV}$. Because the corresponding peaks in the probability density functions of the deposited energy (as illustrated in Figure 6.1) are sharper for muons the effect is much more pronounced.

Muons have a mean life time of $\tau_\mu \simeq 2.2\,\mu\mathrm{s}$. They may decay in the scintillator according to

$$\mu^- \rightarrow \mathrm{e}^- + \nu_\mu + \bar{\nu}_\mathrm{e} \quad \text{or} \quad \mu^+ \rightarrow \mathrm{e}^+ + \bar{\nu}_\mu + \nu_\mathrm{e}.$$

While the neutrinos leave the scintillator the electron or the positron also deposits a part or all of its kinetic energy. Thus, the total energy deposited within the scintillators can be higher than the initial kinetic energy of the muon. Indeed, the plots in Figures 5.7 and 5.8 show a non-zero efficiency for muon energies below the threshold of the upper three energy channels.

Electrons

The electron efficiencies (Figures 5.9 and 5.10) do not show any special characteristics. As for protons and muons about 2% of all electrons lead to a signal in the lowest energy neutron channel.

γ-Rays

As illustrated in Figures 5.11 and 5.12 the efficiencies for γ-rays exhibit no peculiarities except for the channels with anti for the highest zenith angles. Secondary particles produced in high energy photon interactions, i.e. electrons and scattered photons, may cause a veto signal beyond the scintillators leading to a higher veto efficiency.

Comparing the channels with and without anti, respectively, we notice that the veto system reduces the counting rate due to γ-rays only by a factor of ~ 2. Photons can only be detected by the counter tubes if they produce secondary charged particles which cause an ionisation of the counter gas. These secondary particles can be produced either by the photoelectric effect, Compton scattering, or pair production, where the latter is the predominant process for photons above $\sim 10\,\mathrm{MeV}$. Because of the low density of the counter gas, charged particles can only be produced in the metallic mantle. Only a part of the produced electrons reach the counter volume to trigger a pulse. Thus, the response of the counter tubes for γ-rays is much lower than for charged particles.

Comparison with experimental data

Tsuchiya et al. (2001) determined the neutron efficiencies of the Bolivia type detector experimentally. For this purpose they exposed a miniaturized configuration of the detector (Matsubara et al., 1997) perpendicular to a neutron beam. Since they did not use the same threshold energies for the various scintillator channels ($> 30\,\mathrm{MeV}$, $> 60\,\mathrm{MeV}$, $> 90\,\mathrm{MeV}$ and $> 120\,\mathrm{MeV}$) as we used in our simulations, we had to resimulate our detector to obtain comparable information. While the horizontal area of the entire simulated detector is $5.75\,\mathrm{m}^2$, the effective area of the scintillators is only $4\,\mathrm{m}^2$. Therefore, we corrected

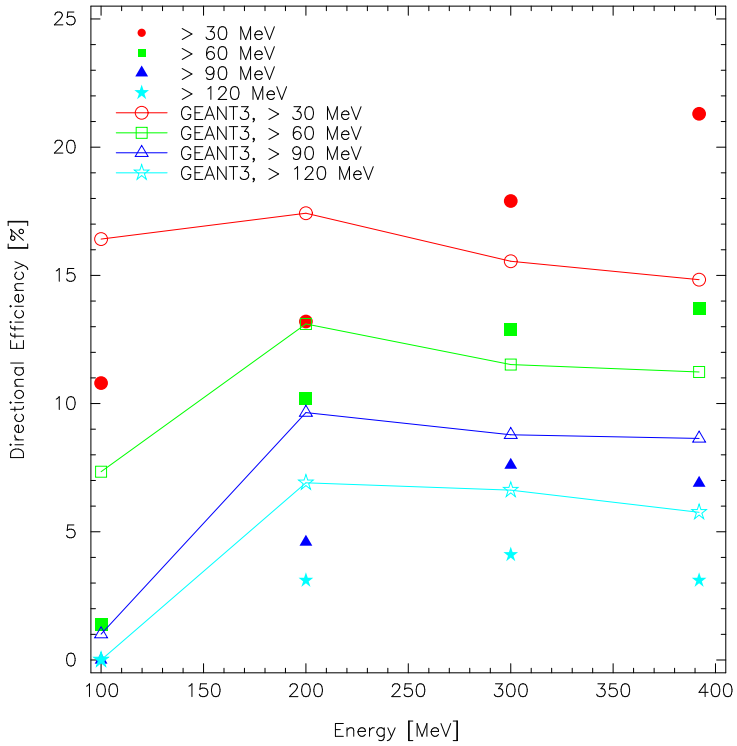


Figure 6.2: Experimental vertical detection efficiency for the Bolivia type detector (open symbols, Tsuchiya et al., 2001) and simulation results (filled symbols) obtained for the Gornergrat detector.

the simulated efficiencies by multiplication with $5.75/4 = 1.44$. These corrected values, together with the experimental data obtained by Tsuchiya et al. (2001), are shown in Figure 6.2. As the plot shows there are significant differences between the data from the accelerator experiment and our simulation results. In general, the simulation overestimates the efficiencies, and the theoretical results exhibit a different energy dependence compared to the experimental values.

The largest overestimation of the detection efficiency occurs for the energy channels $>$ 30 MeV and $>$ 60 MeV at a neutron energy of 100 MeV. A possible reason for this overestimation could be the non-linearity of the relation between the produced light and the deposited energy in the scintillator (Leo, 1987) which is not taken into account in the simulation. The relation between the light output per unit length $\mathrm{d}L/\mathrm{d}x$ and the energy loss $\mathrm{d}E/\mathrm{d}x$ can be approximated by *Birk's law*

$$\frac{\mathrm{d}L}{\mathrm{d}x} = A\frac{\frac{\mathrm{d}E}{\mathrm{d}x}}{1 + kB\frac{\mathrm{d}E}{\mathrm{d}x}}, \tag{6.1}$$

where A is the absolute scintillation efficiency and kB a parameter relating the density of ionization centers to $\mathrm{d}E/\mathrm{d}x$. Therefore, recoil particles with low kinetic energy, i.e. high $\mathrm{d}E/\mathrm{d}x$, have a higher light output than minimum ionizing particles. The highest light output is probably at the bottom of the scintillator material. But for detection by the photomultipliers, light flashes from these positions undergo the strongest attenuation in traversing the thick scintillator. Quantitative estimations of this effect will be subject to further investigations.

For the highest threshold the energy dependence of the theoretical and experimental

efficiencies is somewhat similar. For all other thresholds the theoretical efficiencies show only a slight dependence on energy above 200 MeV, whereas the experimental efficiencies, at least for the two highest thresholds, exhibit a pronounced energy dependence. Only by smoothing out the detection efficiencies shown in the top left panel of Figure 5.3 would the theoretical results show a monotonous increase over the entire energy range up to $\sim 2\,\mathrm{GeV}$. However, the discrepancy could not be fully removed. In addition, this would require a contribution from neutron-pion production processes much different from what is shown in Figure 5.2.

However, we have to consider that the experimental setup was not the same as in our simulations. Tsuchiya et al. (2001) used a spread-up neutron beam covering the entire detector area while we only considered a sharp pencil beam. In addition, the neutron beam was not monoenergetic and the absolute neutron flux could not be estimated accurately.

6.1.2 Directional Counters

Figures 5.19 and 5.23 show that the directional counters have the highest efficiency for neutrons (max. $\sim 10\%$ at 3 GeV) and for γ-rays ($> 10\%$ at > 200 MeV). The neutron efficiency above 3 GeV decreases with increasing energy due to the fact that the lowest energy neutron channel is used as a trigger for the directional counters and that the efficiency of that photomultiplier channel also decreases above 3 GeV (Figure 5.3). The efficiency for γ-rays is high because the anticounter system does not work in an optimal manner to veto photons. After entering the dense scintillator material, high energy photons produce secondary electrons which can penetrate the directional counters and trigger one or multiple signals.

The multiplicity of the directional counting rate caused by neutrons is mainly $m = 1$ for energies up to $\sim 2\,\mathrm{GeV}$ and for a wide range of directions of incidence. Neutrons with higher energies produce a large number of secondary particles with a widespread directional distribution which produce a signal in almost every directional channel ($m = 5$). It can be therefore expected that the reliability in the determination of the direction of incidence decreases with increasing neutron energy. In order to verify this effect we evaluate the relative counting rates of the various directional channels due to neutrons penetrating the detector with different zenith angles in the north-south-plane ($\phi = 0°$). The results are shown in Figure 6.3. The three lower neutron energies correspond to the energies used for the simulation of the June 3, 1982, GLE; the two higher energies were chosen arbitrarily.

It can be clearly seen that the counting rate distributions of the three lower neutron energies are quite similar and would allow for a determination of the direction of incidence. However, the detection efficiency is not maximal for these energies. At higher energies the angular dependence of the different contributions is relatively small. Therefore, the determination of the direction of incidence becomes difficult.

The last plot in Figure 6.3 shows the relative contribution of the directional counting rates according to a simple geometric model of the counter layers without considering interaction processes, i.e. only for particles with straight trajectories. Since the anti counter system is not considered, only the data in the zenith region are comparable to the GEANT3 simulations. Qualitatively, for small zenith angles the simulation results of the lower en-

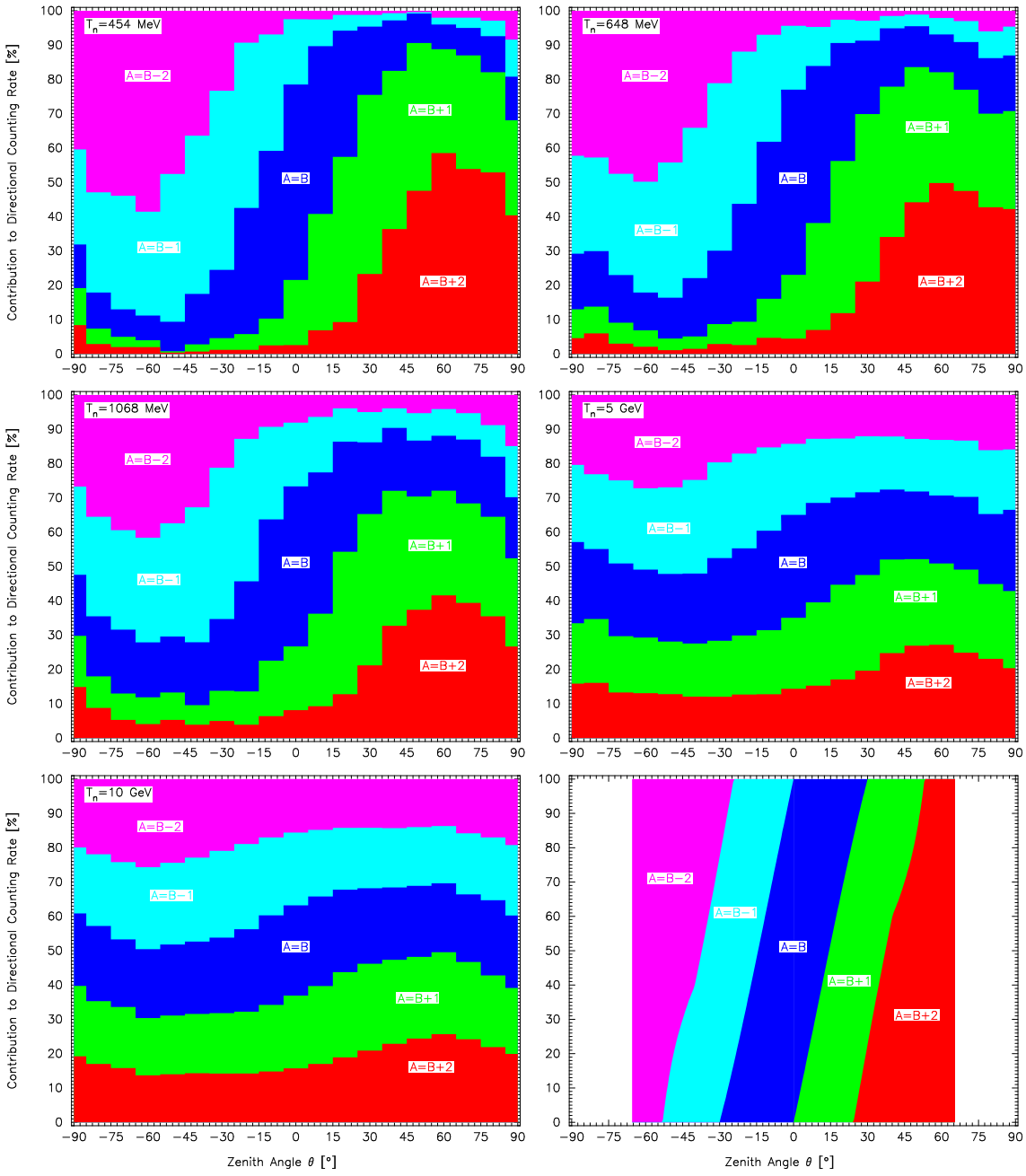


Figure 6.3: Simulated relative contribution to the total directional counting rate for neutrons of different kinetic energy (first 5 pictures) and theoretical contribution obtained from a geometric counter model (last picture).

ergetic particles agree well with the data obtained by the geometric model. It seems that the particle interaction processes flatten the angular dependence only slightly. Due to the multiplicity effect described above, the results for neutrons with energies of 5 and 10 GeV differ essentially from the geometric model, particularly also for small zenith angles.

6.2 Counting Rate due to Galactic Cosmic Rays

If we first compare the simulated total counting rates in Table 5.1 with the experimental counting rates in Table 2.1 we notice with satisfaction that all corresponding values are of

the same order of magnitude. The simulation results however show a tendency to be lower than the corresponding experimental data. The counting rates of the energy channels without anti counters show the best agreement.

Regarding the ratios between the various energy channels, there is a small discrepancy between the theoretical and experimental counting rate ratios of the channels > 80 MeV and > 120 MeV. This effect is due to the enormous difference in the detection efficiency for muons for these two channels as already discussed in Section 6.1.1 and to the fact that muons are the major contributors to the counting rates.

The real total veto counting rate shows the greatest deviation from the theoretical results. For all our simulations we have chosen a lower energy limit of 40 MeV for input particles since lower energy particles cannot be detected by the scintillators. In reality, there is a large flux of particles with energies below this limit, either from cosmic rays or also from radioactive decays in the environment. As these particles are not of interest for the main purpose of SONTEL, we did not make further estimations about their influence on the veto counting rate.

There is also a major difference between the measured and calculated total counting rates without and with anti counters. While the experimental ratio of the counting rates without and with anti has a mean value of ~ 3, the simulated ratio is about 6. In the following we discuss possible reasons for this difference. Some of them concern the simulation itself, others are due to instrumental effects of the detector.

Uncertainties in the input spectra

For several cosmic ray components, especially for neutrons, the uncertainty in the differential spectra from literature is up to $\sim 50\%$. Another source of uncertainty is the approximation of the zenith angle dependence by the $\cos^n$ law (3.2). For example, the uncertainty of the exponent n as given in the literature is about 34% for neutrons and about 5% for muons. Assuming that the intensity of these two cosmic ray components were not correlated, which is actually not true, the simulated ratio of the lowest energy channel would become 5.7 ± 2.4 compared to 2.71 ± 0.04 as obtained from the measurements.

One can try to adapt the weighting factors a_i of (5.4) to fit the theoretical counting rate ratios to the experimental ones. Considering the main contributors to the counting rates, i.e. neutrons and muons, the weighting factors of these components would become $a_{\mathrm{n}} \sim 4$ and $a_\mu \sim 0.7$ for the lowest energy channels. However, such a strong systematic correction of the neutron intensity is inconsistent with the given confidence level of the available neutron and muon spectra. No parameter set a_i could be found to fit the data in accordance with the error bars of all the corresponding input spectra. Thus, the discrepancy in the counting rate ratios cannot be explained by the uncertainty of the input spectra.

Uncertainties in the discriminator settings

Since it is not trivial to correctly choose the discriminator voltage corresponding to the energy levels 40, 80, 120 and 160 MeV, we tried to evaluate the effect of different discriminator level settings. The SONTEL data seem to show transient environmental effects, probably due to radioactive rainout. As the energy of the corresponding radiation is

smaller than the nominal discriminator energies, we evaluated the total counting rates using two different sets of discriminator energies, i.e. 30, 60, 90, 120 MeV and 20, 40, 60, 80 MeV, respectively. For this evaluation again only the main contributors were considered and the corresponding simulations were made with reduced accuracy. The results are shown in Table 6.1.

Detection channel		> 40 MeV > 80 MeV > 120 MeV > 160 MeV	> 30 MeV > 60 MeV > 90 MeV > 120 MeV	> 20 MeV > 40 MeV > 60 MeV > 80 MeV
ch1	with anti	3003	3152	3526
ch2		1697	1991	2385
ch3		813	1184	1793
ch4		450	683	1347
ch1	without anti	17135	17718	18527
ch2		12190	14493	15904
ch3		4221	8997	13817
ch4		1755	3846	11097
A = B − 2	with anti	139	137	140
A = B − 1		267	267	272
A = B		516	519	527
A = B + 1		267	267	272
A = B + 2		139	137	140

Table 6.1: Simulated galactic counting rates [cts/10 s] of all detection channels. Each column represents another set of discriminator levels. Only the main contributors of the counting rates were resimulated.

It can be seen that the counting rates increase with decreasing discriminator levels and that the relative increase in the counting rates due to lower discriminator levels is much higher for the higher energy channels than for the lower ones. The ratios between the counting rates without and with anti allow a clear conclusion. While the counting rate ratio of the lowest energy channel without and with anti decreases only slightly from 5.7 to 5.3, the ratio of the highest channels increases from 3.9 to 8.2. This latter tendency is completely inconsistent with the experimental value of 3.2. Furthermore, the counting rate ratios between the highest and the lowest energy channel also increase, increasing the discrepancy with the observations.

We therefore conclude that if the actual discriminator levels of SONTEL were lower than assumed, the counting rate ratios between the higher and the lower energy channels should be higher than observed. Since the simulation results for the discriminator energies 40, 80, 120 and 160 MeV provide the best agreement with the observed ratios, it seems unlikely that the main discrepancy of our simulation is based on values of the discriminator levels that are too low.

Shortcomings in the electronics

The most probable reason for the discrepancy between the theoretical and experimental results concerning the scintillator channels without and with anti is incorrect tuning of the signal processing electronics. If the veto signal does not coincide with the scintillator signal within the specified range of time (see Figure 2.11), a count in the channel with anti is registered by mistake. Thus, the counting rate of the channels with anti would be too high, in concurrence with observations.

There is another indication that our veto shield may not be as efficient as it should be. The Solar Neutron Telescope in Tibet (4300 m a.s.l.) has almost the same configuration as the Gornergrat detector but with a larger effective area (9 m^2 instead of 4 m^2). However, it only has a veto shield at its northern and southern sides implicating a counting rate of the neutron channels that is too high. Nevertheless, the counting rate ratio between the energy channel > 40 MeV without and with anti is ~ 4, much closer to but still higher than the experimental value from Gornergrat. Of course, this rough comparison neglects the different altitudes and cut-off rigidities of the two detector locations.

In order to identify possible shortcomings in the electronics, it will be necessary to perform extended checks of the signal processing electronics of SONTEL.

The simulated total counting rates of the directional counters are also significantly lower than the experimental values. This is in accordance with the simulated trigger channel (> 40 MeV with anti) that also has a lower counting rate than observed. In order to obtain the counting rates we integrated for particles incident from 6 different zenith angles. Due to the complex characteristics of the directional counting rates as e.g. shown in Figure 6.3, the zenith angles implicated by the condition $\Delta \cos\theta = \text{const.}$ are probably not optimal for simulating the directional counts. Because we chose $\Delta \cos\theta = \text{const.}$ instead of $\Delta\theta = \text{const.}$ the simulated directions have rather high zenith angles. However, at high zenith angles deviations from the cosine law (3.2) are most pronounced.

The simulation results also yield detailed information about the contributions of the different species of radiation to the various counting rates as illustrated in Table 6.2. Neutrons, muons, and γ-rays are the main contributors to the counting rates of the energy channels with anti as well as to the directional counting rates. The counting rates without anti are dominated by the muon component (76% in ch1).

Radiation type	SONTEL > 40 MeV		SONTEL	IGY	NM64
	with anti	without anti	A = B		
neutron	40%	8%	29%	84%	85%
proton	2%	4%	2%	7%	6%
muon	30%	76%	40%	7%	6%
electron	2%	6%	1%		
γ-ray	26%	6%	27%		

Table 6.2: Contributions of the various radiation types to the total counting rates of SONTEL at Gornergrat (> 40 MeV, A = B) and of an IGY and NM64 neutron monitor at high latitude sea-level locations (Hatton, 1971).

Table 6.2 also shows a comparison of the relative contributions between SONTEL at Gornergrat and IGY and NM64 neutron monitors at high latitude sea-level stations (Hatton, 1971). While the neutron monitors are mainly sensitive to the nucleonic component of the secondary cosmic radiation ($> 90\%$), the neutron channels of SONTEL also contain large contributions by muons and γ-rays which together e.g. amount up to 50% in the > 40 MeV channel. Cosmic ray muons are also the main contributor to the counting rates without anti and to the veto counting rate. As a consequence the counting rates without anti and the veto counting rate are expected to show a dependence on atmospheric effects as e.g. on the temperature profile in the atmosphere (see e.g. Dorman, 1974).

6.3 Atmospheric Cascade

6.3.1 Solar Neutron Event of June 3, 1982

In the past, several attempts have been made to derive the secondary neutron spectra due to primary solar neutrons. Shibata (1994) developed a Monte Carlo method using nuclear interaction models including both elastic and inelastic scattering processes. As an illustration of the Shibata results, Figure 6.4 shows the simulated omnidirectional energy spectra of neutrons at an atmospheric depth of $776\,\mathrm{g\,cm^{-2}}$ and for different primary energies $T_\mathrm{n} = 150$, 200, 300, 500, 700 and 900 MeV.

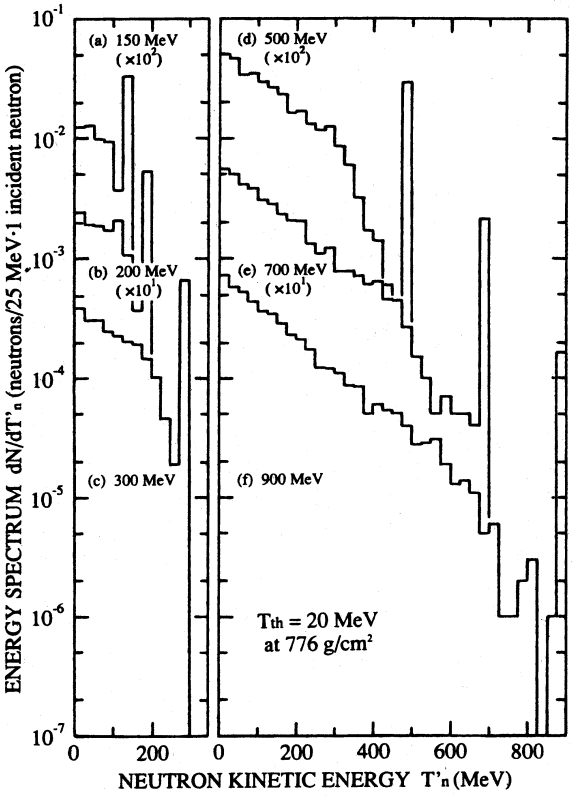


Figure 6.4: The energy spectra of neutrons at an atmospheric depth of $776\,\mathrm{g\,cm^{-2}}$. The histograms correspond to the following neutron energies at the top of the atmosphere: (a) $T_\mathrm{n} = 150\,\mathrm{MeV}$, (b) 200 MeV, (c) 300 MeV, (d) 500 MeV, (e) 700 MeV, and (f) 900 MeV (from Shibata, 1994).

The energy spectra can be characterized by two main features: a narrow peak close to the primary neutron energy and a continuum extending over the whole energy range below. The narrow peak corresponds to primary neutrons which penetrate through the Earth's atmosphere without any collision or which undergo only marginal elastic scattering (*punch-through*). The continuum is assumed to arise from inelastically scattered primary neutrons and from neutrons produced in the atmosphere.

In order to compare the results from Shibata (1994) with our simulations, we concentrate on the attenuation of the neutrons, i.e. the average number of neutrons reaching the observational level in an atmospheric cascade induced by a primary neutron. The discussion is made separately for the punch-through component and for the component of inelastic scattering. We have normalized the simulation results of Shibata (1994) to Gornergrat altitude by multiplying by a factor of 2.0, taken from Figure 5 in the same paper. Table 6.3 shows the comparison with our GEANT3 simulations.

T_n	Attenuation	Shibata (1994)		Our results	
648 MeV	Punch-through	4.4×10^{-4}	(6.3%)	4.4×10^{-4}	(0.6%)
	Inelastic scattering	6.6×10^{-3}	(93.7%)	7.3×10^{-2}	(99.4%)
	Total	7.0×10^{-3}		7.3×10^{-2}	
454 MeV	Punch-through	7.0×10^{-4}	(11.3%)	1.1×10^{-3}	(2.2%)
	Inelastic scattering	5.5×10^{-3}	(88.7%)	5.1×10^{-2}	(97.8%)
	Total	6.2×10^{-3}		5.2×10^{-2}	

Table 6.3: Comparison of the neutron attenuation in the atmosphere at $700\,\mathrm{g\,cm^{-2}}$ between the model of Shibata (1994) and our results. The table indicates the attenuation by no or marginal elastic scattering (punch-through), by inelastic scattering, and their sum. The relative contributions of the two scattering types are indicated in parentheses.

The comparison of the punch-through attenuations shows that the simulation results agree within a factor of 2. On the other hand, the attenuations of the component of inelastic scattering differ by one order of magnitude. Our simulation results are higher than the values obtained by Shibata (1994). This difference is probably due to large uncertainties in the inelastic cross sections and differences in the nuclear interaction models used in various simulation codes. In contrast to Shibata our results suggest that in a solar neutron event the contribution of punch-through neutrons to the total number of neutrons arriving at Gornergrat altitude is relatively small (2.2% at 454 MeV).

We also derived the secondary spectra of protons, muons, electrons, and γ-rays induced by a flux of primary neutrons. The intensity of these components is at least one order of magnitude smaller than the corresponding neutron intensity, except for low energy γ-rays. Due to the high sensitivity of SONTEL to this component, γ-rays will contribute significantly to the response of the detector to solar neutrons.

6.3.2 Solar Proton Event of April 15, 2001

The secondary particle spectra due to primary solar protons show a different behavior than the secondary spectra resulting from primary solar neutrons. There is no radiation type dominating the entire energy range under consideration. Below $\sim 100\,\mathrm{MeV}$, γ-rays

are dominant. From $\sim 100\,\mathrm{MeV}$ up to $\sim 1\,\mathrm{GeV}$ neutrons are the most abundant particle species, competing at higher energies with protons and muons. Because every radiation type contributes significantly to the total differential flux, it seems justified that all of them are included in the simulation of the detector response to solar protons.

6.4 Detector Response for Solar Particles

6.4.1 Solar Neutron Event of June 3, 1982

Since the Solar Neutron Telescope at Gornergrat has probably not yet identified any solar neutron event, we compare the simulation results with neutron monitor data of the June 3, 1982, GLE. Figure 6.5 shows the relative 1-minute counting rate of the 18-IGY neutron monitor at Jungfraujoch (Chupp et al., 1987). The simulated relative increases of the SONTEL energy channels with anti are also indicated for comparison.

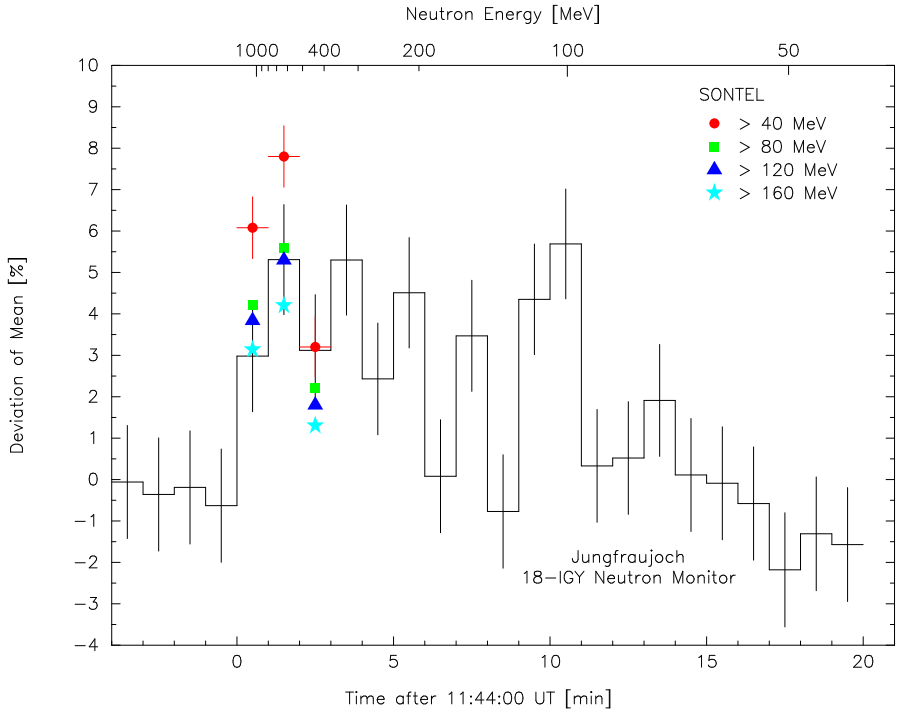


Figure 6.5: Increase in the 18-IGY neutron monitor counting rate at Jungfraujoch during the solar neutron event on June 3, 1982 (Chupp et al., 1987). The simulated increases in the SONTEL neutron channels are indicated for neutron energies 1068, 648, and 454 MeV.

Because the first solar neutrons reached the IGY neutron monitor at about 11:44:22 UT (Debrunner et al., 1983) and contributed only to a part of the sampling interval, the increase in the 1-minute average counting rate is reduced. For the comparison, we corrected the first increase of the simulated SONTEL counting rates accordingly.

The lowest energy channel of SONTEL shows a significantly higher response to solar neutrons of energies 1068 and 648 MeV than the IGY neutron monitor, whereas the response to 454 MeV neutrons is almost equal for both detectors. The relative increase in the counting rates of the higher energy channels is only about 40–70% of the lowest energy channel. Simulations for lower energy neutrons corresponding to the counting

rates after 11:47:00 UT did not show any significant increase. Thus, SONTEL shows only a response to the highest energy solar neutrons, i.e. to primary neutrons with energies above ~ 400 MeV, at least for neutron events similar to the June 3, 1982, event.

We conclude that the SONTEL neutron channels would respond to a solar neutron event such as the June 3, 1982, GLE, during the first three minutes by a counting rate increase of ~ 4–$10\sigma_{1\mathrm{min}}$ in the lowest energy channel and only a maximum of $\sim 2\sigma_{1\mathrm{min}}$ in the highest energy channel. The Jungfraujoch IGY neutron monitor showed an increase of maximum $5\sigma_{1\mathrm{min}}$, but it showed significant increased counting rates even after 11:47:00 UT. This may be due to a time extended instead of a δ-like neutron injection.

Our simulation of the June 3, 1982, solar neutron event shows that the neutron component is the main contributor to all SONTEL counting rates (Table 5.3), whereas the influence of other radiation is minor. If we calculate the ratios between the increase in the lowest energy channel with and without anti for the three primary energies, we obtain values > 0.8, which is indicative for an increased neutron flux. This conclusion is independent of the fact that our simulations did not properly reproduce the experimental counting rate ratios between the antied and the non-antied energy channels, since that discrepancy is a consequence of the detection properties of the veto counters and therefore is due to charged particles and γ-rays. The simulation of the atmospheric cascade of solar neutrons showed that the intensity increase in the charged particle and γ-ray component is only of secondary order.

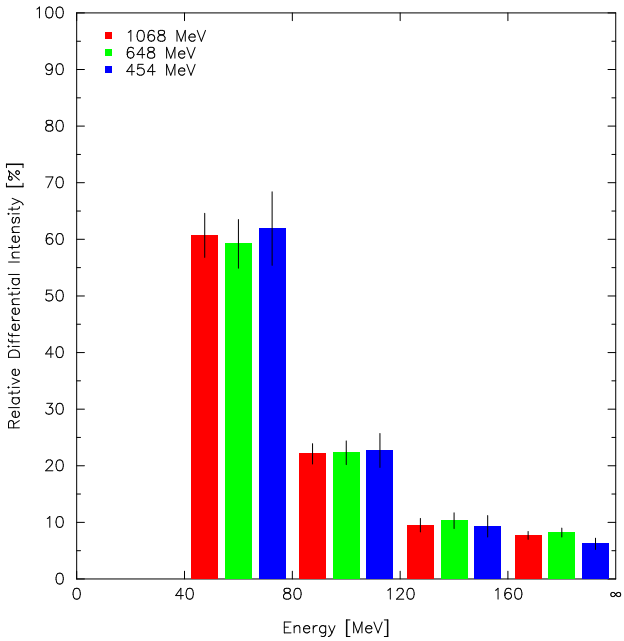


Figure 6.6: Simulated differential energy spectrum of the neutron channels. The values refer to the primary neutron energies 1068, 648, and 454 MeV, indicated in relative units of the integral counting rate > 40 MeV.

We already pointed out the importance of the energy determination by the detector. Evaluating the 1-minute counting rates of the four energy channels at the three selected primary energies, we find no statistical evidence for a variation in the shape among the three differential spectra, as shown in Figure 6.6. Therefore, it is not possible to derive information about the primary kinetic energy of neutrons penetrating the detector, at

least at energies where SONTEL is sensitive, i.e. above $\sim 400\,\mathrm{MeV}$. Estimates on the primary neutron energy still have to be made via the arrival time delay according to (5.5) under the assumption of a δ-like injection, or based on more comprehensive information about the time extended solar neutron emission.

The increases in all the directional counting rates due to solar neutrons are only marginal and are statistically not significant ($\Delta N \lesssim 0.5\sigma_{1\mathrm{min}}$). Therefore, the results do not allow to derive information on the primary angle of incidence.

Since we simulated the solar neutron response of the detector for three different energies, it is possible to obtain the corresponding values of the *solar neutron sensitivity function* $S_\mathrm{n}(X, T_0, \theta_0)$. For each detection channel this sensitivity function is defined (Debrunner et al., 1997) via the absolute counting rate increase

$$\Delta N(X, t, \theta_0) = \int \left(\frac{\mathrm{d}J_\mathrm{n}}{\mathrm{d}T}\right)_{T_0,t} \cdot\, S_\mathrm{n}(X, T_0, \theta_0)\,\mathrm{d}T_0, \tag{6.2}$$

where X is the atmospheric depth, T_0 and θ_0 the energy and the zenith angle of the primary neutron, and $\mathrm{d}J_\mathrm{n}/\mathrm{d}T$ the differential energy spectrum of the primary neutron flux at the time t. For discrete neutron energies (6.2) can be rewritten as

$$S_\mathrm{n}(X, T_0, \theta_0) = \frac{\Delta N(X, T_0, \theta_0)}{J_\mathrm{n}(T_0)}. \tag{6.3}$$

For the various detection channels, Table 6.4 lists the derived values of S_n for primary neutrons with zenith angle $\theta_0 = 25.5°$. The sensitivities of an 18-IGY neutron monitor

Detection channel		$S_\mathrm{n}(X = 700\,\mathrm{g\,cm^{-2}}, T_0, \theta_0 = 25.5°)$ [m^2]		
		$T_0 = 1068\,\mathrm{MeV}$	$T_0 = 648\,\mathrm{MeV}$	$T_0 = 454\,\mathrm{MeV}$
> 40 MeV	with anti	2.9×10^{-2}	1.3×10^{-2}	8.6×10^{-3}
> 80 MeV		1.2×10^{-2}	5.2×10^{-3}	3.4×10^{-3}
> 120 MeV		5.0×10^{-3}	2.3×10^{-3}	1.4×10^{-3}
> 160 MeV		2.2×10^{-3}	1.1×10^{-3}	5.3×10^{-4}
> 40 MeV	without anti	3.5×10^{-2}	1.4×10^{-2}	9.4×10^{-3}
> 80 MeV		1.5×10^{-2}	6.4×10^{-3}	3.8×10^{-3}
> 120 MeV		7.6×10^{-3}	3.2×10^{-3}	1.7×10^{-3}
> 160 MeV		4.1×10^{-3}	1.6×10^{-3}	7.1×10^{-4}
A = B − 2	with anti	1.0×10^{-4}		
A = B − 1		2.0×10^{-4}	5.5×10^{-5}	
A = B		4.1×10^{-4}	1.6×10^{-4}	8.8×10^{-5}
A = B + 1		2.0×10^{-4}	5.5×10^{-5}	
A = B + 2		1.0×10^{-4}		
18-IGY		5.6×10^{-3}	3.8×10^{-3}	3.0×10^{-3}
6-NM64		4.5×10^{-3}	3.3×10^{-3}	2.7×10^{-3}

Table 6.4: Sensitivities of the SONTEL detection channels to solar neutrons ($\theta_0 = 25.5°$). The results are given for the three primary neutron energies T_0 =1068, 648, and 454 MeV. The last two rows show the sensitivities of a 18-IGY and a 6-NM64 neutron monitor, respectively.

(Chupp et al., 1987; Debrunner et al., 1997) and of a 6-NM64 (Stein, 1989) at $700\,\mathrm{g\,cm^{-2}}$ are indicated for comparison. It can be seen that at the three simulated neutron energies, the SONTEL sensitivities for the two lower energy neutron channels are higher than the sensitivities of the two neutron monitor types, e.g. by a factor of $\gtrsim 3$ at > 40 MeV. As the average SONTEL counting rate of the lowest energy neutron channel ($\sim 48000\,\mathrm{min^{-1}}$) is more than five times higher than the average IGY counting rate ($\sim 8800\,\mathrm{min^{-1}}$), the relative increases in a solar neutron event are almost equal for both SONTEL and the neutron monitor (e.g. Figure 6.5). For the two higher energy channels the SONTEL sensitivities are lower than the neutron monitor sensitivities. The sensitivities of the directional channels are much lower than those of the energy channels, e.g. up to two orders of magnitude between the lowest energy channel and the directional channel A = B (zenith).

6.4.2 Solar Proton Event of April 15, 2001

During the operation period of SONTEL at Gornergrat (since 1998) a single GLE was recorded on April 15, 2001. Figure 6.7 shows the relative counting rates of the lowest energy neutron channel and of the veto counters of SONTEL, as well as of the Jungfraujoch IGY neutron monitor during the same event.

The data both from Gornergrat and Jungfraujoch show an onset time at $\sim$ 14:00 UT. The lowest energy neutron channel of SONTEL had a maximum increase in the counting rate of $\sim$ 2% around 14:28 UT while the neutron monitor reached a maximum increase of $\sim$ 8% around 14:18 UT. This figure also includes environmental radioactivity data at Gornergrat obtained with a *GammaTRACER* (Genitron Instruments GmbH) counter tube.

Because the simulated mean counting rates of the neutron channels differ from the experimental data (as discussed in Section 5.2), we concentrate on the absolute rather than on the relative increases of the GLE counting rates. The simulated and the experimental absolute SONTEL counting rate increases are given in Table 6.5. All values refer to 14:17 UT.

Detection channel		ΔN [min^{-1}]	
		simulated	experimental
> 40 MeV		270 ± 16	1626 ± 284
> 80 MeV	with	144 ± 12	1038 ± 221
> 120 MeV	anti	72 ± 8	396 ± 131
> 160 MeV		42 ± 6	120 ± 57
> 40 MeV		558 ± 24	2190 ± 359
> 80 MeV	without	348 ± 19	1554 ± 290
> 120 MeV	anti	108 ± 10	630 ± 170
> 160 MeV		54 ± 7	138 ± 76

Table 6.5: Simulated and experimental absolute SONTEL counting rate increases during the April 15, 2001, GLE at 14:17 UT.

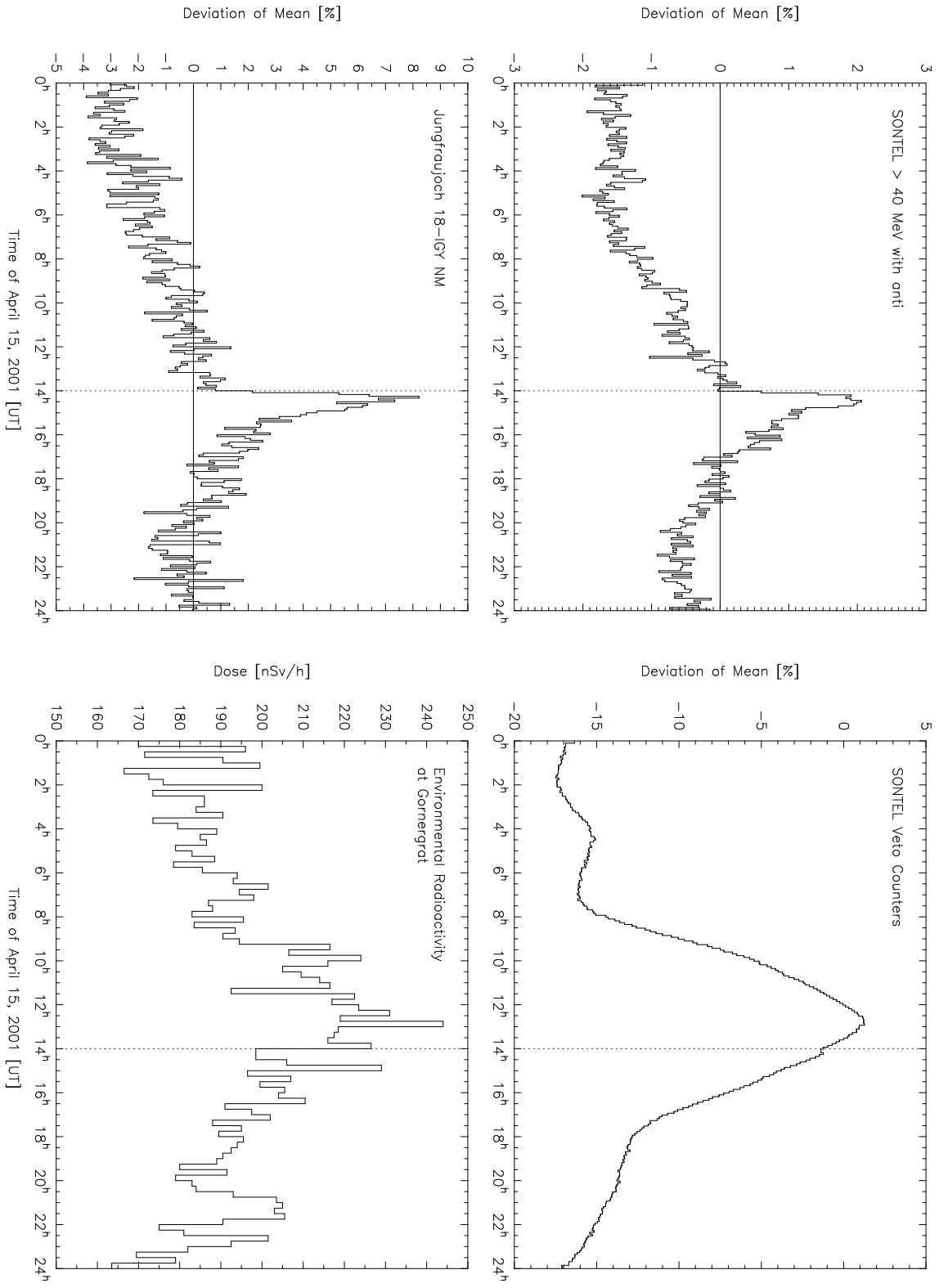


Figure 6.7: Relative 5-minute counting rates of the lowest energy neutron channel and the veto counters of SONTEL and of the 18-IGY neutron monitor at Jungfraujoch during the April 15, 2001, GLE. The environmental radioactivity at Gornergrat is indicated as 15-minute average values.

The comparison between the detection channels with and without anti shows that the simulated counting rate ratio is smaller than the experimental one. This is probably due to the fact that the electronic circuits of SONTEL were not tuned optimally during that period of time. After a later adjustment of the electronics the counting rates with anti were reduced by a factor of $\sim 2/3$. Assuming that the increase in the counting rates of the neutron channels would also be reduced by this factor, the simulated and the experimental mean ratios between the counting rates with and without anti would become quite similar (~ 0.5). We notice that for the solar neutron event the corresponding ratio was higher (> 0.8) since neutrons were the dominating radiation. Thus, the ratio between the counting rates with and without anti seems to be a valuable measure to distinguish between a solar neutron and a solar proton event.

Comparing the absolute increases ΔN between the simulation and the SONTEL data we find that, in the mean, the simulated values are by a factor of ~ 4 lower than the experimental data. We assume that this difference results mainly from the uncertainty in the solar proton spectrum (Lockwood et al., 2002), which is estimated by the authors to be within a factor of two. Furthermore, we supposed in our simulations that the protons hit the atmosphere with zenith angles up to 30°. A maximum zenith angle of 40° would increase the resulting particle fluxes by a factor of ~ 2. Also the use of a simplified model of the Earth's atmosphere in our simulation may contribute to the difference.

We also compare the simulated relative differential energy spectra with the experimental data. Figure 6.8 shows the differential spectra of the energy channels with anti (left panel) and without anti (right panel) for the four energy intervals covered by SONTEL.

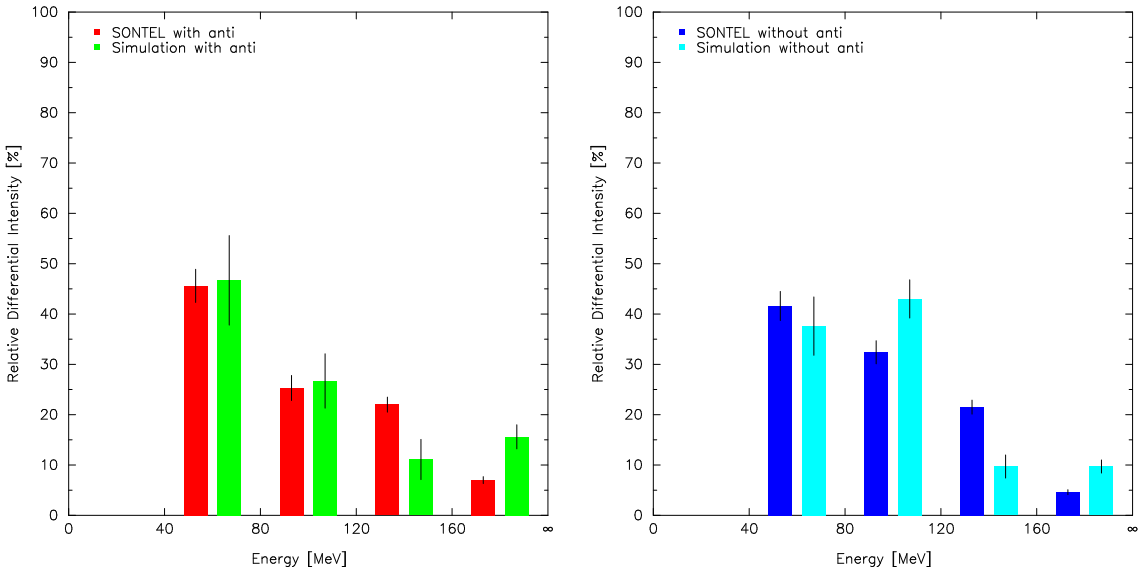


Figure 6.8: Experimental and simulated relative differential energy spectra of the energy channels with anti (left) and without anti (right) for the April 15, 2001, GLE at 14:17 UT.

The simulated differential spectra agree with SONTEL data at a 1σ-confidence level in the energy interval 40–80 MeV with and without anti and in the energy interval 80–120 MeV with anti. The simulation has a tendency to overestimate the counting rates > 160 MeV. This behavior is also assumed to be a consequence of the uncertainty in the primary and, therefore, in the simulated secondary particle spectra. However, the energy spectra ob-

tained for the solar neutron event of June 3, 1982, (Figure 6.6) are significantly softer than the spectrum at 14:17 UT during the solar proton event of April 15, 2001, in accordance with observations. Thus, the shape of the differential spectrum can also be used to distinguish between the two types of events.

The veto counting rate during April 15, 2001, plotted in the top right panel of Figure 6.7, shows an enormous increase starting about six hours before the onset of the GLE. Simultaneously, an increase in the environmental radioactivity was observed at Gornergrat. Since the energy channels do not show such an effect, the energy of the radiation causing the veto increase must have been lower than 40 MeV. At present we do not have any explanation of the origin of this low energy radiation. Most probably the effect has a terrestrial cause since, as far as we know, no space detector registered an associated signal.

Finally, we note that the simulated increase in the directional counting rates during the GLE showed no statistically significant increase, thus in agreement with the experimental data.

Chapter 7

Summary

In order to correctly interpret the measurements of the Solar Neutron Telescope (SONTEL) at Gornergrat, in particular during solar neutron events, detailed knowledge of the atmospheric cosmic ray cascade and of the detector properties is necessary. In this work we present, after a technical description of SONTEL, three GEANT3 Monte Carlo applications, which we developed to improve this knowledge. From a first simulation we obtained the efficiencies of the various SONTEL detection channels for neutrons, protons, muons, electrons, and γ-rays in the energy range $40\,\mathrm{MeV} \leq T \leq 20\,\mathrm{GeV}$. Using these efficiencies and particle spectra adapted from literature, we then evaluated average SONTEL counting rates caused by secondary cosmic rays. The results are discussed in an extensive comparison with observations. In a second application we simulated the atmospheric cascades of solar neutrons and protons during the June 3, 1982, solar neutron event and the April 15, 2001, solar proton event to obtain the corresponding secondary particle spectra above the detector. A standard atmosphere was used for this application. The results allowed a comparison with estimates made by other authors. In the last application we calculated the SONTEL detector response to solar neutrons and protons using the input spectra obtained from the second application. The results are discussed in comparison with data from the Jungfraujoch neutron monitors and SONTEL.

The simulation results of the mean counting rates show that neutrons and, in contrast to neutron monitors, muons and γ-rays are the main contributors to the counting rates of the SONTEL neutron channels. Muons are the dominating contributors to the counting rates without anti and to the veto counts. While the theoretical counting rates without anti agree well with the experiment, the simulated counting rates with anti are smaller by a factor of ~ 2. Shortcomings in the detector veto electronics are assumed to be responsible for an excessively high counting rate of the neutron channels. The calculated directional counting rates also differ from the observed ones. This difference could probably be diminished by another choice of simulation parameters.

The evaluation of the simulation results of the atmospheric cascades induced by solar neutrons showed that the relative amount of primary solar neutrons arriving at observational level is lower by one order of magnitude than assumed for the design of the detector. The simulation of the June 3, 1982, solar neutron event showed that the lowest energy neutron channel of SONTEL can identify solar neutrons above $\sim 400\,\mathrm{MeV}$ with a significance of up to $\sim 10\sigma$ in the 1-minute values. The simulation also revealed that the

measured differential energy spectra do not provide significant information on the primary neutron spectrum. Also due to the low efficiency of the directional counters, the minor increase in the corresponding counting rates cannot be used to derive the direction of incident neutrons. The comparison of the simulation results for the April 15, 2001, solar proton event with measurements served as a further verification of the applications. Regarding the uncertainty in the primary cosmic ray particle spectrum used as input for the simulation, we found that the theoretical and experimental results agree satisfactorily. As a noteworthy feature we found that for the two simulated events, the counting rate ratio between the SONTEL energy channels with and without anti provide a clear distinction between primary solar neutrons and protons. Compared with neutron monitor data from Jungfraujoch, SONTEL was less sensitive to the solar proton event by a factor of ~ 4.

The relevance of this work includes the following aspects: First, the summary of the technical details within the first part of this work will be used in the future as an operator's manual. Second, based on the simulation, significant new insight in the response of SONTEL to primary and secondary cosmic ray particles was obtained. Third, the developed applications provide a convenient tool to simulate possible improvements of SONTEL such as the optimal choice of the discriminator levels or the installation of additional absorbers to improve the directional resolution. Furthermore, they can be used for the development of new ground based solar neutron detectors, since our GEANT3 routines allow an easy adaption to other altitudes, detector geometries, and material parameters. Finally, the GEANT4 applications which were also developed during this work will provide the state of the art basis for future Monte Carlo studies in this domain, as soon as improved hadronic interaction models become available.

Acknowledgments

First of all, I would like to express sincere thanks to Prof. Erwin Flückiger. By entrusting me with the ambitious topic of this diploma thesis he allowed my dealing with a very interesting and fascinating field of experimental physics. As an exemplary supervisor he always followed the progress of my work with great interest, and he provided me with advice at any time.

This work would not have been feasible without the assistance of Dr. Laurent Desorgher. Owing to his long experience in the field he succeeded in introducing me carefully to the philosophy and the practical use of GEANT4. He also supported me by translating the Monte Carlo applications to GEANT3. I am very grateful for his encouragement.

I also appreciate the support of Dr. Rolf Bütikofer, with whom I shared the office during my work. I could profit by his detailed knowledge about the SONTEL detector either during visits at Gornergrat or in many inspirational dialogs at the Institute. Furthermore, he let me have parts of his technical documentation as well as several plots for use within this thesis.

For her professional English language support I am very grateful to our secretary, Mrs. Louise Wilson. She also considerably contributed to the pleasant working environment within the Cosmic Ray Group.

Finally, sincere thanks are given to all the other persons who supported me and my work (not only) over recent years!

This work was kindly supported by the Swiss National Science Foundation (grants NF 20-057175.99 and NF 20-067092.01).

Bibliography

Allkofer, O. C. (1971). *Teilchen-Detektoren*. Verlag Karl Thiemig KG, München, Germany.

Allkofer, O. C. and P. K. F. Grieder (1984). *Physics Data: Cosmic Rays on Earth*. Fachinformationszentrum Energie Physik Mathematik GmbH Karlsruhe, Germany.

Biermann, L., O. Haxel, and A. Schlüter (1951). Neutrale Ultrastrahlung von der Sonne. *Z. Naturforsch. 6a*, 47.

Bütikofer, R. (1988). Zum Wiederaufbau des Multiplizitätsneutronenmonitors in Bern. Master's thesis, Physikalisches Institut, Universität Bern, Switzerland.

Bütikofer, R., E. O. Flückiger, Y. Muraki, et al. (2001). The upgraded solar neutron detector at Gornergrat. In *Proc. 27th Int. Cosmic Ray Conf., Hamburg, Germany*, pp. 3053–3055.

CERN (1993). *GEANT — Detector Description and Simulation Tool*. Application Software Group, Computing and Networks Division, CERN Geneva, Switzerland. Available on the CERNLIB WWW pages (URL: http://wwwinfo.cern.ch/asdoc/Welcome.html).

Chilingarian, A. (2002). Aragats Space-Environmental Center (ASEC): Status and SEP Forecasting Possibilities. In Y. Muraki (Ed.), *Proc. 22nd ISTC Japan Workshop on Space Weather Forecast in Russia/CIS*, Volume 2, pp. 45–71.

Chupp, E. L., H. Debrunner, E. Flückiger, et al. (1987). Solar neutron emissivity during the large flare on 1982 June 3. *Ap. J. 318*, 913–925.

Chupp, E. L., D. J. Forrest, J. M. Ryan, et al. (1982). A Direct Observation of Solar Neutrons Following the 0118 UT Flare on 1980 June 21. *Ap. J. (Lett.) 263*, L95.

Debrunner, H., E. Flückiger, E. L. Chupp, et al. (1983). The Solar Cosmic Ray Neutron Event on June 3, 1982. In *Proc. 18th Int. Cosmic Ray Conf., Bangalore, India*, Volume 4, pp. 74–78.

Debrunner, H., J. A. Lockwood, C. Barat, et al. (1997). Energetic Neutrons, Protons and Gamma Rays during the 1990 May 24 Solar Cosmic-Ray Event. *Ap. J. 479*, 997–1011.

Dorman, L. I. (1974). *Cosmic Rays: Variations and Space Explorations*. North-Holland Publishing Company, Amsterdam, The Netherlands.

Flückiger, E. O. (1976). *Ein Programmsystem zur Beschreibung der Wechselwirkung zwischen hochenergetischen Nukleonen und Materie und seine Anwendung bei kosmischen Strahlungsuntersuchungen.* Physikalisches Institut, Universität Bern, Switzerland.

Flückiger, E. O. (1977). Theoretical spectra of cosmic ray neutrons in the atmosphere for the energy range 50 MeV $\leq E \leq$ 100 GeV. In *Proc. 15th Int. Cosmic Ray Conf., Plovdiv, Bulgaria*, Volume 4, pp. 144.

Flückiger, E. O., R. Bütikofer, Y. Muraki, et al. (1998). A new solar neutron telescope at Gornergrat. In J. Medina (Ed.), *rayos cósmicos 98. Proc. 16th European Cosmic Ray Symposium, Departamento de Fisica, Universidad de Alcalá, Spain*, pp. 219–222.

Grieder, P. K. F. (2001). *Cosmic rays at earth: researcher's reference manual and data book.* Elsevier Science B.V., Amsterdam, The Netherlands.

Groom, D. E., M. Aguilar-Benitez, C. Amsler, et al. (2000). Review of Particle Physics. *The European Physical Journal C15*(1–4), 1–878. Available on the PDG WWW pages (URL: http://pdg.lbl.gov/).

Hatton, C. J. (1971). The neutron monitor. In J. G. Wilson and S. A. Wouthuysen (Eds.), *Progress in Elementary Particle and Cosmic Ray Physics*, Volume X, pp. 1–100. North-Holland Publishing Company, Amsterdam, The Netherlands.

Klein, H. J. and J. Zoll (1988). *PATCHY Reference Manual,* Program Library L400. CERN Geneva, Switzerland.

Leo, W. R. (1987). *Techniques for Nuclear and Particle Physics Experiments, A How-to Approach.* Springer-Verlag, Berlin, Germany.

Lide, D. R. (Ed.) (1998). *CRC Handbook of Chemistry and Physics* (79 ed.). National Institute of Standards & Technology, USA.

Lockwood, J. A., H. Debrunner, E. O. Flückiger, and J. M. Ryan (2002). Solar proton rigidity spectra from 1 to 10 GV of selected flare events since 1960. *Solar Physics 208*(1), 113–140.

Lockwood, J. A., H. Debrunner, and J. M. Ryan (1997). The relationship between solar flare gamma-ray emission and neutron production. *Solar Physics 173*, 151–176.

Longair, M. S. (1992). *High Energy Astrophysics; Particles, photons and their detection*, Volume 1. Cambridge University Press, Cambridge, U.K.

Matsubara, Y., Y. Muraki, T. Sako, et al. (1997). Detection Efficiency of the Bolivian Solar Neutron Detector. In *Proc. 25th Int. Cosmic Ray Conf., Durban, South Africa*, Volume 1, pp. 61–64.

Muraki, Y., T. Takahashi, Y. Matsubara, et al. (1993). The properties of Norikura solar neutron telescope # 2. In *Proc. 23th Int. Cosmic Ray Conf., Calgary, Canada*, Volume 3, pp. 171–174.

Pyle, K. R. and J. A. Simpson (1991). Observation of a direct solar neutron event on 22 March 1991 with the Haleakala, Hawaii, neutron monitor. In *Proc. 22th Int. Cosmic Ray Conf., Dublin, Ireland*, Volume 3, pp. 53–56.

Ryan, J., D. Forrest, J. Lockwood, et al. (1993). COMPTEL Gamma Ray and Neutron Measurements of Solar Flares. In *AIP Conf. Proc. 280, St. Louis, MO, USA*, pp. 631–642.

Share, G. H., P. L. Nolan, D. J. Forrest, et al. (1982). Measurements of the 2.223 MeV Neutron Capture Line in Solar Flares. *Bull. AAS 14*, 875.

Shibata, S. (1994). Propagation of solar neutrons through the atmosphere of the Earth. *J. Geophys. Res. 99*(A4), 6651–6665.

Shibata, S., K. Murakami, Y. Muraki, et al. (1991). The detection efficency of a new solar neutron telescope. In *Proc. 22th Int. Cosmic Ray Conf., Dublin, Ireland*, Volume 3, pp. 788–791.

Stein, P. (1989). Zur Empfindlichkeit des NM64-Neutronenmonitors für solare Neutronen. Master's thesis, Physikalisches Institut, Universität Bern, Switzerland.

Tsuchiya, H. (2001). *Observation of solar neutrons in association with solar flare using an international network of new type of solar neutron detector.* Ph. D. thesis, Solar-Terrestrial Environment Laboratory, Nagoya University, Nagoya, Japan.

Tsuchiya, H., Y. Matsubara, Y. Muraki, et al. (2001). Observations of neutrons in association with the solar flare of 6 November 1997. In *Proc. 27th Int. Cosmic Ray Conf., Hamburg, Germany*, pp. 3040–3043.

Tsuchiya, H., Y. Muraki, K. Masuda, et al. (2001). Detection efficency of a new type of solar neutron detector calibrated by an accelerator neutron beam. *Nucl. Instr. and Meth. A 463*, 183–193.

Appendix A

Source Code of GEANT3 Applications

A.1 Application SONTEL

A.1.1 main.F

```
C=====================================================================C
C MAIN.F - Part of the SONTEL GEANT3 application                       C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/26/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      PROGRAM SONTELB
C---------------------------------------------------------------------C
C     Detector simulation with rotating source and zenith angle       C
C     dependence of intensity                                         C
C=====================================================================C

#include "geant321/gcflag.inc"
#include "sontel.inc"

      COMMON/PAWC/H(1000000)
      COMMON/GCBANK/Q(5000000)

      INTEGER NRTHETA, NRPHI
      REAL ZENITHINT, COSEXP
      REAL DELTACOS, DELTAPHI, OMEGA, TIME, AREA
      REAL COSTHETA, THETA, PHI
      REAL PI
      REAL RINTENSITY(9,9)
      DATA RINTENSITY/81*0/
      PARAMETER(PI=3.1415926535)
      CHARACTER FILE*4, RUNNAME*5

      CALL TIMEST(1E5)
      CALL GZEBRA(5000000)
      CALL HLIMIT(-1000000)
      CALL UGINIT
```

```
C === Source definition ===

      DELTACOS=1./(NRTHETA-1)
      DELTAPHI=PI/2./(NRPHI-1)
      OMEGA=DELTACOS*DELTAPHI
      AREA=SORADIUS**2.*PI

C --- Intensity ---

      DO M=0,NRTHETA-1,1
        DO N=0,NRPHI-1,1
          COSTHETA=1.-M*DELTACOS
         PHI=N*DELTAPHI
         IF (M.LT.NRTHETA-1) THEN
         RINTENSITY(M+1,N+1)=ZENITHINT*
     *     OMEGA*COSTHETA**COSEXP
          ELSE
           RINTENSITY(M+1,N+1)=0.
         ENDIF
        ENDDO
      ENDDO

C --- Starts the runs for various positions ---

      DO M=0,NRTHETA-1,1
        DO N=0,NRPHI-1,1
         COSTHETA=1.-M*DELTACOS
         THETA=ACOS(COSTHETA)*180./PI
         PHI=N*DELTAPHI*180./PI
          IF ((N.EQ.0).OR.(M.GT.0.AND.M.LT.NRTHETA-1)) THEN
            CALL RESETHISTO
          ENDIF
          IF ((M.EQ.0.AND.N.EQ.0).OR.(M.GT.0.AND.M.LT.NRTHETA-1)) THEN
            CALL USERRUN(THETA,PHI,INT(RINTENSITY(M+1,N+1)*AREA*TIME))
          ENDIF
          WRITE(FILE,200)M,N
200       FORMAT('_',I1,'_',I1)
          WRITE(RUNNAME,300)IFILENR
300       FORMAT(I5)
          CALL STOREHISTO(RUNNAME//FILE)
        ENDDO
      ENDDO

      CALL UGLAST

      END


C=====================================================================C
      SUBROUTINE USERRUN(THETA,PHI,NOEVENTS)
C---------------------------------------------------------------------C
C     Positioning of the source and start of the simulation with a    C
C     certain number of events to be produced                         C
C=====================================================================C

#include "sontel.inc"

      COMMON/PRODEVENTS/NPRODEVENTS
      INTEGER NOEVENTS

      NPRODEVENTS=0
      IF (ISOURCEON.NE.0) THEN
        CALL NEW_SOURCE_ORIENTATION(PHI,THETA)
      ENDIF
```

```
100   CALL GTRIGI
      CALL GTRIG
      CALL GTRIGC
      IF (NPRODEVENTS.LT.NOEVENTS) GOTO 100

      END


C=====================================================================C
      SUBROUTINE STOREHISTO(FILENAME)
C---------------------------------------------------------------------C
C     Stores histograms                                               C
C=====================================================================C

      CHARACTER FILENAME*9, FILEEXT*6, DIR*6

      DIR='hbook/'
      FILEEXT='.hbook'
      CALL HRPUT(0,DIR//FILENAME//FILEEXT,'T')

      END


C=====================================================================C
      SUBROUTINE RESETHISTO
C---------------------------------------------------------------------C
C     Resets histograms                                               C
C=====================================================================C

      DO I=1,16
        CALL HRESET(I,' ')
      ENDDO

      END
```

A.1.2 main2.F

```
C=====================================================================C
C MAIN2.F - Part of the SONTEL GEANT3 application                      C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/26/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      PROGRAM SONTELEFF
C---------------------------------------------------------------------C
C     Detector simulation with rotating source and constant intensity   C
C=====================================================================C

#include "geant321/gcflag.inc"
#include "sontel.inc"

      COMMON/PAWC/H(1000000)
      COMMON/GCBANK/Q(5000000)

      INTEGER NRTHETA, NRPHI
      REAL ZENITHINT, COSEXP
      REAL DELTACOS, DELTAPHI, OMEGA, TIME, AREA
      REAL COSTHETA, THETA, PHI
      REAL PI
      REAL RINTENSITY(9,9)
      DATA RINTENSITY/81*0/
      PARAMETER(PI=3.1415926535)
      CHARACTER FILE*4, RUNNAME*5

      CALL TIMEST(1E5)
      CALL GZEBRA(5000000)
      CALL HLIMIT(-1000000)
      CALL UGINIT

C === Source definition ===

      DELTACOS=1./(NRTHETA-1)
      DELTAPHI=PI/2./(NRPHI-1)
      OMEGA=DELTACOS*DELTAPHI
      AREA=SORADIUS**2.*PI

C --- Intensity ---

      DO M=0,NRTHETA-1,1
        DO N=0,NRPHI-1,1
          COSTHETA=1.-M*DELTACOS
         PHI=N*DELTAPHI
         RINTENSITY(M+1,N+1)=ZENITHINT*OMEGA
        ENDDO
      ENDDO

C --- Starts the runs for various positions ---

      DO M=0,NRTHETA-1,1
        DO N=0,NRPHI-1,1
         COSTHETA=1.-M*DELTACOS
         THETA=ACOS(COSTHETA)*180./PI
         PHI=N*DELTAPHI*180./PI
          IF ((N.EQ.0).OR.(M.GT.0.AND.M.LT.NRTHETA)) THEN
            CALL RESETHISTO
          ENDIF
```

```
          IF ((M.EQ.0.AND.N.EQ.0).OR.(M.GT.0.AND.M.LT.NRTHETA)) THEN
            CALL USERRUN(THETA,PHI,INT(RINTENSITY(M+1,N+1)*AREA*TIME))
          ENDIF
          WRITE(FILE,200)M,N
200       FORMAT('_',I1,'_',I1)
          WRITE(RUNNAME,300)IFILENR
300       FORMAT(I5)
          CALL STOREHISTO(RUNNAME//FILE)
        ENDDO
      ENDDO

      CALL UGLAST

      END


C=====================================================================C
      SUBROUTINE USERRUN(THETA,PHI,NOEVENTS)
C---------------------------------------------------------------------C
C     Positioning of the source and start of the simulation with a    C
C     certain number of events to be produced                         C
C=====================================================================C

#include "sontel.inc"

      COMMON/PRODEVENTS/NPRODEVENTS
      INTEGER NOEVENTS

      NPRODEVENTS=0
      IF (ISOURCEON.NE.0) THEN
        CALL NEW_SOURCE_ORIENTATION(PHI,THETA)
      ENDIF
100   CALL GTRIGI
      CALL GTRIG
      CALL GTRIGC
      IF (NPRODEVENTS.LT.NOEVENTS) GOTO 100

      END

C=====================================================================C
      SUBROUTINE STOREHISTO(FILENAME)
C---------------------------------------------------------------------C
C     Stores histograms                                               C
C=====================================================================C

      CHARACTER FILENAME*9, FILEEXT*6, DIR*6

      DIR='hbook/'
      FILEEXT='.hbook'
      CALL HRPUT(0,DIR//FILENAME//FILEEXT,'T')

      END

C=====================================================================C
      SUBROUTINE RESETHISTO
C---------------------------------------------------------------------C
C     Resets histograms                                               C
C=====================================================================C

      DO I=1,16
        CALL HRESET(I,' ')
      ENDDO

      END
```

A.1.3 guhadr.F

```
C=====================================================================C
C GUHADR.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

      SUBROUTINE GUHADR

      CALL GFMFIN

      END
```

A.1.4 gukine.F

```
C=====================================================================C
C GUKINE.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE GUKINE
C---------------------------------------------------------------------C
C     Generates Kinematics for primary track                          C
C=====================================================================C

#include "geant321/gcbank.inc"
#include "geant321/gcflag.inc"
#include "geant321/gckine.inc"
#include "geant321/gconsp.inc"
#include "geant321/gcscan.inc"
#include "celoss.inc"
#include "pvolum.inc"
#include "sontel.inc"

      CHARACTER*20 CHNPAR
      DIMENSION VERTEX(3),PLAB(3),RNDM(1),UB(10)
      DIMENSION NID1(1000),NID2(10000),NID3(10000)
      DIMENSION NID4(10000),NID5(36000),SDIR(3)
      SAVE NID1, NID2,NID3, NID4,NID5
      DATA NID1/1000*0/NID2/10000*0/NID3/10000*0/NID4/10000*0/
      DATA NID5/36000*0/
      SAVE VERTEX,PLAB
      DATA VERTEX/0.,1.,70./
      DATA PLAB  /3*0./
      COMMON/PRODEVENTS/NPRODEVENTS

      IF(SCANFL) THEN
         CALL GSCANK
      ELSE
         DEDL=0.
         IF (ISPECTRUM.NE.0) THEN
          EKIN=COMPUTE_ENERGY(SELEKIN,SELSPECDIFF,SELSPECINT,NSEL)
         ELSE
          IF(IKINE.EQ.0) THEN
           EKIN=PKINE(1)
          ELSE
           CALL GRNDM(RNDM,1)
           EKIN=PKINE(1)+(PKINE(2)-PKINE(1))*RNDM(1)
          ENDIF
         ENDIF
         CALL GFPART(IPARTICLE,CHNPAR,ITRTYP,AMASS,CHARGE,TLIFE,UB,NWB)
         ETOT=EKIN+AMASS
         PLAB(3)=SQRT((ETOT+AMASS)*(ETOT-AMASS))
         IF (ISOURCEON.EQ.0) THEN
          VERTEX(1)=POSSTART(1)
          VERTEX(2)=POSSTART(2)
          VERTEX(3)=POSSTART(3)
          fac=3.1415926/180.
          phi=PHISTART*fac
          theta=THETASTART*fac
          PLAB(1)=PLAB(3)*SIN(theta)*COS(phi)
          PLAB(2)=PLAB(3)*SIN(theta)*SIN(phi)
          PLAB(3)=PLAB(3)*COS(theta)
```

```
         ELSE
   36     CALL CIRCULAR_SOURCE(VERTEX,SDIR)
          PLAB(1)=PLAB(3)*SDIR(1)
          PLAB(2)=PLAB(3)*SDIR(2)
          PLAB(3)=PLAB(3)*SDIR(3)
          fac=3.1415926/180.
          phi=sophi*fac
          theta=sotheta*fac
          ICONDITION=1
          IF (ICHECKCROSSING.NE.0)
     *        ICONDITION=ICROSSING_WITH_DETECTOR(VERTEX,SDIR)
          IF (ICONDITION.EQ.0) THEN
            IEVENT=IEVENT+1
            NPRODEVENTS=NPRODEVENTS+1
            GOTO 36
          ENDIF
         ENDIF
         NPRODEVENTS=NPRODEVENTS+1
         CALL GSVERT(VERTEX,0,0,0,0,NVERT)
         CALL GSKINE(PLAB,IPARTICLE,NVERT,0,0,NT)

C --- register primary particle ---
         EKININC=EKIN*1000.
         CALL HFF1(1,NID1,EKIN*1000.,1.)
         CALL HFF2(2,NID2,VERTEX(1),VERTEX(2),1.)
         CALL HFF2(3,NID3,VERTEX(2),VERTEX(3),1.)
         CALL HFF2(4,NID4,VERTEX(3),VERTEX(1),1.)
         CALL HFF2(5,NID5,phi/fac,cos(theta),1.)

C --- initialisation GUSTEP ---
         IVETO=1
         IDIR0SN=1
         IDIR25S=1
         IDIR25N=1
         IDIR40S=1
         IDIR40N=1
         IDIR(1)=1
         IDIR(2)=1
         IDIR(3)=1
         IDIR(4)=1
         IDIR(5)=1

         DO I=1,124
          ETOTCOUNTER(I) = 0
         ENDDO
          ETOTCOUNTERS = 0
         DO I=1,4
          ETOTSCINT(I) = 0
         ENDDO
         ETOTSCINTS = 0
         ICH1=1
         ICH2=1
         ICH3=1
         ICH4=1

C --- kinematics debug (controlled by ISWIT(1)) ---
         IF(IDEBUG.EQ.1) THEN
            IF(ISWIT(1).EQ.1) THEN
               CALL GPRINT('VERT',0)
               CALL GPRINT('KINE',0)
            ENDIF
         ENDIF
      ENDIF
      END
```

```
C=====================================================================C
      SUBROUTINE ELECTRON_INTSPEC
C---------------------------------------------------------------------C
C     Normalizes integral and differential electron spectrum          C
C=====================================================================C

#include "sontel.inc"

      DIMENSION  DATAEKIN(15), DATASPEC(15)

      DATA DATAEKIN/ 3.930900e+01, 6.485400e+01, 9.993200e+01,
     *    1.764600e+02, 2.976100e+02, 4.688800e+02, 7.906300e+02,
     *    1.136800e+03, 1.749200e+03, 2.570800e+03, 3.778600e+03,
     *    5.187700e+03, 7.453400e+03, 1.071000e+04, 2.065600e+04/

      DATA DATASPEC/ 7.806600e-05, 3.622500e-05, 1.836900e-05,
     * 6.938100e-06,2.546340e-06, 1.082850e-06, 3.748200e-07,
     * 1.643160e-07,
     *    5.694300e-08, 1.917420e-08, 6.648300e-09, 2.594010e-09,
     *   9.264600e-10, 3.407100e-10, 5.339400e-11/

      EMIN=40.
      EMAX=20000.
      NELE=15
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,ELEEKIN,ELESPECDIFF,
     *                     ELESPECINT,NELE, EMIN,EMAX)
      NSEL=15
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,SELEKIN,SELSPECDIFF,
     *                     SELSPECINT,NSEL, EMIN,EMAX)
      END


C=====================================================================C
      SUBROUTINE NEUTRON_INTSPEC
C---------------------------------------------------------------------C
C     Normalizes integral and differential neutron spectrum (za)      C
C=====================================================================C

#include "sontel.inc"

      DIMENSION  DATAEKIN(18), DATASPEC(18)

      DATA DATAEKIN/40.,75., 150., 300., 500., 700., 900., 1125., 1375.,
     *               1625., 1875., 2500., 4000.,  6.250000e+03,
     *      8.750000e+03, 1.250000e+04,1.750000e+04 , 2.500000e+04/

      DATA DATASPEC/1.399999e-05,1.400000e-05,1.600000e-05,
     * 8.600000e-06,
     * 3.900000e-06,2.400000e-06, 1.600000e-06,1.200000e-06,
     * 8.600000e-07, 6.700000e-07, 5.400000e-07,5.100000e-07,
     * 1.800000e-07, 5.700000e-08, 2.100000e-08,5.400000e-09 ,
     * 2.800000e-09, 7.400000e-10 /

      EMIN=40.
      EMAX=20000.
      NNEU=18
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,NEUEKIN,NEUSPECDIFF,
     *                     NEUSPECINT,NNEU, EMIN,EMAX)
      NSEL=18
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,SELEKIN,SELSPECDIFF,
     *                     SELSPECINT,NSEL, EMIN,EMAX)
      END
```

```
C======================================================================C
      SUBROUTINE NEUTRON2_INTSPEC
C----------------------------------------------------------------------C
C     Normalizes integral and differential neutron spectrum (fl)       C
C======================================================================C

#include "sontel.inc"

      DIMENSION  DATAEKIN(18), DATASPEC(18)

      DATA DATAEKIN/40.,75., 150., 300., 500., 700., 900., 1125., 1375.,
     *               1625., 1875., 2500., 4000.,  6.250000e+03,
     *      8.750000e+03, 1.250000e+04,1.750000e+04 , 2.500000e+04/

      DATA DATASPEC/5.599999e-05,5.600000e-05,1.600000e-05,
     * 8.600000e-06,
     * 3.900000e-06,2.400000e-06, 1.600000e-06,1.200000e-06,
     * 8.600000e-07, 6.700000e-07, 5.400000e-07,5.100000e-07,
     * 1.800000e-07, 5.700000e-08, 2.100000e-08,5.400000e-09 ,
     * 2.800000e-09, 7.400000e-10 /

      EMIN=40.
      EMAX=20000.
      NNEU2=18
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,NEUEKIN2,NEUSPECDIFF2,
     *                     NEUSPECINT2,NNEU2, EMIN,EMAX)
      NSEL=18
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,SELEKIN,SELSPECDIFF,
     *                     SELSPECINT,NSEL, EMIN,EMAX)
      END

C======================================================================C
      SUBROUTINE MUON_INTSPEC
C----------------------------------------------------------------------C
C     Normalizes integral and differential muon spectrum               C
C======================================================================C

#include "sontel.inc"

      DIMENSION  DATAEKIN(20), DATASPEC(20)

      DATA DATAEKIN/ 4.000000e+01, 5.704571e+01, 9.448304e+01,
     * 1.479395e+02, 2.169248e+02, 3.311225e+02, 4.505481e+02,
     * 6.235752e+02, 7.408699e+02, 9.545184e+02, 1.459510e+03,
     * 1.883149e+03, 2.621188e+03, 3.877742e+03,
     * 5.088615e+03, 6.928934e+03, 9.422926e+03,  1.231779e+04,
     * 1.759365e+04, 2.297458e+04/

      DATA DATASPEC/ 1.431585e-03, 2.081260e-03, 3.161337e-03,
     * 4.323975e-03, 5.414375e-03, 5.747502e-03, 5.588054e-03,
     * 4.706667e-03, 4.106715e-03, 3.477111e-03, 2.158524e-03,
     * 1.580132e-03, 1.012861e-03, 5.519642e-04, 3.540833e-04,
     * 1.959821e-04, 1.185073e-04, 6.371031e-05, 3.275058e-05,
     * 1.923420e-05/

      EMIN=40.
      EMAX=20000.
      NMUO=20
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,MUOEKIN,MUOSPECDIFF,
     *                     MUOSPECINT,NMUO, EMIN,EMAX)
      NSEL=20
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,SELEKIN,SELSPECDIFF,
     *                     SELSPECINT,NSEL, EMIN,EMAX)
      END
```

```
C=====================================================================C
      SUBROUTINE PROTON_INTSPEC
C---------------------------------------------------------------------C
C     Normalizes integral and differential proton spectrum            C
C=====================================================================C

#include "sontel.inc"

      DIMENSION  DATAEKIN(18), DATASPEC(18)

      DATA DATAEKIN/0.,75., 150., 300., 500., 700., 900., 1125., 1375.,
     *              1625., 1875., 2500., 4000.,  6.250000e+03,
     *      8.750000e+03, 1.250000e+04,1.750000e+04 , 2.500000e+04/

      DATA DATASPEC/4.399999e-06, 4.400000e-06, 4.000000e-06,
     * 2.700000e-06, 2.000000e-06, 1.500000e-06, 1.300000e-06,
     * 7.500000e-07, 7.200000e-07, 6.000000e-07, 5.100000e-07,
     * 4.500000e-07, 1.600000e-07, 5.700000e-08, 1.700000e-08,
     * 6.000000e-09, 3.500000e-09, 1.500000e-09 /

      EMIN=40.
      EMAX=20000.
      NPRO=18
      NSEL=18
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,PROEKIN,PROSPECDIFF,
     *                     PROSPECINT,NPRO, EMIN,EMAX)
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,SELEKIN,SELSPECDIFF,
     *                     SELSPECINT,NSEL, EMIN,EMAX)
      END


C=====================================================================C
      SUBROUTINE GAMMA_INTSPEC
C---------------------------------------------------------------------C
C     Normalizes integral and differential gamma spectrum             C
C=====================================================================C

#include "sontel.inc"

      DIMENSION  DATAEKIN(15), DATASPEC(15)

      DATA DATAEKIN/4.000000e+01, 8.139500e+01,  1.256500e+02,
     * 1.725000e+02, 2.460500e+02, 3.441700e+02, 6.217000e+02,
     * 8.869700e+02, 1.317700e+03, 1.919100e+03,
     * 2.581400e+03, 3.915200e+03, 7.373400e+03, 1.507200e+04,
     * 2.069600e+04/

      DATA DATASPEC/1.652600e-04, 6.718600e-05,3.001200e-05,
     * 1.278000e-05, 6.739400e-06,3.553700e-06, 1.410400e-06,
     * 7.092300e-07,  2.619900e-07,1.039100e-07, 5.350200e-08 ,
     * 1.522200e-08,  3.337800e-09, 3.114500e-10, 1.264800e-10/

      EMIN=40.
      EMAX=20000.
      NGAM=15
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,GAMEKIN,GAMSPECDIFF,
     *                     GAMSPECINT,NGAM, EMIN,EMAX)
      NSEL=15
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,SELEKIN,SELSPECDIFF,
     *                     SELSPECINT,NSEL, EMIN,EMAX)
      END
```

```
C=====================================================================C
      FUNCTION COMPUTE_ENERGY(EK,SPECDIFF,SPECINT,NPOINT)
C---------------------------------------------------------------------C
C     Computes initial energy                                         C
C=====================================================================C

      DIMENSION RNDM(1),EK(50), SPECDIFF(50), SPECINT(50)

      CALL GRNDM(RNDM,1)
      DO i=2, NPOINT
        IF ((RNDM(1).GE.SPECINT(i-1))
     *                  .AND.(RNDM(1).LE.SPECINT(i))) THEN
        x0=EK(i-1)
        x1=EK(i)
        A0=SPECDIFF(i-1)
        A1=SPECDIFF(i)
        REST=RNDM(1)-SPECINT(i-1)
        Ax=REST*ALOG(A1/A0)/(x1-x0) + A0
        EKIN=((x1-x0)*ALOG(Ax/A0)/ALOG(A1/A0)) + x0
        COMPUTE_ENERGY=EKIN/1000.
        RETURN
        ENDIF
      ENDDO
      END


C=====================================================================C
      SUBROUTINE COMPUTE_INTSPEC(DATAEK,DATASPEC,EK,SPECDIFF,
     *                                  SPECINT,NPOINT, EMIN,EMAX)
C---------------------------------------------------------------------C
C     Computes integral spectrum                                      C
C=====================================================================C

#include "sontel.inc"

      DIMENSION  DATAEK(50), DATASPEC(50), EK(50), SPECDIFF(50)
      DIMENSION  SPECINT(50)

      EK(1)=EMIN
      x=EMIN
      x0=DATAEK(1)
      x1=DATAEK(2)
      A0=DATASPEC(1)
      A1=DATASPEC(2)
      SPECDIFF(1)=A0*( (A1/A0) ** ((x-x0)/(x1-x0)) )
      EK(NPOINT)=EMAX
      x=EMAX
      x0=DATAEK(NPOINT-1)
      x1=DATAEK(NPOINT)
      A0=DATASPEC(NPOINT-1)
      A1=DATASPEC(NPOINT)
      SPECDIFF(NPOINT)=A0*( (A1/A0) ** ((x-x0)/(x1-x0)) )
      DO i=2, NPOINT-1
        SPECDIFF(i)=DATASPEC(i)
        EK(i)=DATAEK(i)
      ENDDO
      SPECINT(1)=0.
      DO i=2, NPOINT
        x0=EK(i-1)
        x1=EK(i)
        A0=SPECDIFF(i-1)
        A1=SPECDIFF(i)
        SPECINT(i)=SPECINT(i-1)+ ((x1-x0)*(A1-A0)/ALOG(A1/A0))
      ENDDO
```

```
      DO i=1, NPOINT
       SPECDIFF(i)=SPECDIFF(i)/SPECINT(NPOINT)
       SPECINT(i)=SPECINT(i)/SPECINT(NPOINT)
      ENDDO
      END



C=====================================================================C
      SUBROUTINE CIRCULAR_SOURCE(POS,DIR)
C---------------------------------------------------------------------C
C     Computes position distribution of the circular source           C
C=====================================================================C

#include "sontel.inc"

      DIMENSION POS(3),DIR(3),RNDM(1)

      CALL GRNDM(RNDM,1)
      radius=SORADIUS*SQRT(RNDM(1))
      CALL GRNDM(RNDM,1)
      phi=3.1415926*2.*RNDM(1)
      COSPHI=COS(phi)
      SINPHI=SIN(phi)
      DO i=1,3
       POS(I)=SOCENTER(I)+SODISTANCE*SDIRR(I)
     *              +RADIUS*(COSPHI*SDIRTH(I)+SINPHI*SDIRPH(I))
       DIR(I)=-SDIRR(I)
      ENDDO
      END



C=====================================================================C
      SUBROUTINE NEW_SOURCE_ORIENTATION(PHI,THETA)
C---------------------------------------------------------------------C
C     Computes orientation of the circular source                     C
C=====================================================================C

#include "sontel.inc"

      SOPHI=PHI
      SOTHETA=THETA
      DEGRAD=3.1415926/180.
      COSPHI=COS(PHI*DEGRAD)
      SINPHI=SIN(PHI*DEGRAD)
      COSTH=COS(THETA*DEGRAD)
      SINTH=SIN(THETA*DEGRAD)
      SDIRR(1)=COSPHI*SINTH
      SDIRR(2)=SINPHI*SINTH
      SDIRR(3)=COSTH
      SDIRTH(1)=COSPHI*COSTH
      SDIRTH(2)=SINPHI*COSTH
      SDIRTH(3)=-SINTH
      SDIRPH(1)=-SINPHI
      SDIRPH(2)=COSPHI
      SDIRPH(3)=0
      END
```

```
C=======================================================================C
      FUNCTION ICROSSING_WITH_DETECTOR(POS,DIR)
C-----------------------------------------------------------------------C
C     Checks if the particle will cross the detector                    C
C=======================================================================C

#include "sontel.inc"

      DIMENSION POS(3),DIR(3)

      ICROSSING_WITH_DETECTOR=0
      PREC=0.0000000000001
      TMIN=-1000./PREC
      TMAX=1000./PREC
      DO I=1,3
       P=POS(I)-DETCENTER(I)
      IF (ABS(DIR(I)).GE.PREC) THEN
       T1=(-DETSIZE(I)/2. - P)/DIR(I)
       T2=(DETSIZE(I)/2. - P)/DIR(I)
       TPMIN=MIN(T1,T2)
       TPMAX=MAX(T1,T2)
       TMIN=MAX(TMIN,TPMIN)
       TMAX=MIN(TMAX,TPMAX)
       IF (TMIN.GT.TMAX) RETURN
      ELSE
       IF (ABS(P).GT.(DETSIZE(I)/2.)) RETURN
      ENDIF
      ENDDO
      ICROSSING_WITH_DETECTOR=1
      RETURN
      END
```

A.1.5 guout.F

```
C==========================================================================C
C GUOUT.F - Part of the SONTEL GEANT3 application                          C
C--------------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                             C
C Last modification: 08/16/2002                                            C
C==========================================================================C

#include "geant321/pilot.h"

C==========================================================================C
      SUBROUTINE GUOUT
C--------------------------------------------------------------------------C
C     User routine called at the end of each event                         C
C==========================================================================C

#include "geant321/gcflag.inc"
#include "geant321/gcscan.inc"
#include "pvolum.inc"
#include "celoss.inc"
#include "sontel.inc"

      DIMENSION NID6(2),NID7(5),NID8(5)
      DIMENSION NID9(2000),NID10(4),NID11(4)
      DIMENSION NID12(5000),NID13(5000),NID14(5),NID15(5000)
      DIMENSION NID16(5000)

      SAVE NID6, NID7,NID8, NID9,NID10,NID11
      DATA NID6/2*0/NID7/5*0/NID8/5*0/NID9/2000*0/
      DATA NID10/4*0/NID11/4*0/
      DATA NID12/5000*0/NID13/5000*0/NID14/5*0/NID15/5000*0/
      DATA NID16/5000*0/

C --- Scintillators ---
      DO I=1,4
       IF (ETOTSCINT(I).GT.DISC(1)) THEN
        ICH1=2
        IF (ETOTSCINT(I).GT.DISC(2)) THEN
         ICH2=2
         IF (ETOTSCINT(I).GT.DISC(3)) THEN
          ICH3=2
          IF (ETOTSCINT(I).GT.DISC(4)) ICH4=2
         ENDIF
        ENDIF
       ENDIF
      ENDDO
      IF (ICH1.EQ.2) THEN
       CALL HFF1(11,NID11,1.,1.)
       IF (IVETO.EQ.1) CALL HFF1(10,NID10,1.,1.)
       CALL HFF2(16,NID16,EKININC,1.,1.)
       IF (IVETO.EQ.1) CALL HFF2(15,NID15,EKININC,1.,1.)
      ENDIF
      IF (ICH2.EQ.2) THEN
       CALL HFF1(11,NID11,2.,1.)
       IF (IVETO.EQ.1) CALL HFF1(10,NID10,2.,1.)
       CALL HFF2(16,NID16,EKININC,2.,1.)
       IF (IVETO.EQ.1) CALL HFF2(15,NID15,EKININC,2.,1.)
      ENDIF
      IF (ICH3.EQ.2) THEN
       CALL HFF1(11,NID11,3.,1.)
       IF (IVETO.EQ.1) CALL HFF1(10,NID10,3.,1.)
       CALL HFF2(16,NID16,EKININC,3.,1.)
       IF (IVETO.EQ.1) CALL HFF2(15,NID15,EKININC,3.,1.)
      ENDIF
```

```
      IF (ICH4.EQ.2) THEN
       CALL HFF1(11,NID11,4.,1.)
       IF (IVETO.EQ.1) CALL HFF1(10,NID10,4.,1.)
       CALL HFF2(16,NID16,EKININC,4.,1.)
       IF (IVETO.EQ.1) CALL HFF2(15,NID15,EKININC,4.,1.)
      ENDIF
      IF ( ETOTSCINTS.GT.0.) THEN
        CALL HFF1(9,NID9,ETOTSCINTS,1.)
      ENDIF

C --- Directional counters ---

      DO I=86,124,2
          IF (ETOTCOUNTER(I).GT.EIONI) THEN
           J1=(MAX(85,I-5)+1-I)/2+3
           J2=(MIN(123,I+3)+1-I)/2+3
           DO J=J1,J2
            IF (ETOTCOUNTER(I+(J-3)*2-1).GT.EIONI) THEN
             IDIR(J)=2
            ENDIF
           ENDDO
          ENDIF
      ENDDO
      NDETDIR=0
      DO I=1,5
      IF (IDIR(I).EQ.2) THEN
       VAL=I-3.
       CALL HFF1(8,NID8,VAL,1.)
       IF (IVETO.EQ.1.AND.ICH1.EQ.2) CALL HFF1(7,NID7,VAL,1.)
       NDETDIR=NDETDIR+1
      ENDIF
      ENDDO
      IF (NDETDIR.NE.0) THEN
      IF (IVETO.EQ.1.AND.ICH1.EQ.2)
     *   CALL HFF2(12,NID12,EKININC,1.*NDETDIR,1.)
      CALL HFF2(13,NID13,EKININC,1.*NDETDIR,1.)
      CALL HFF1(14,NID14,1.*NDETDIR,1.)
      ENDIF
      CALL HFF1(6,NID6,IVETO-1.,1.)
      END
```

A.1.6 guphad.F

```
C=====================================================================C
C GUPHAD.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

      SUBROUTINE GUPHAD

      CALL GFMDIS

      END
```

A.1.7 gustep.F

```
C=======================================================================C
C GUSTEP.F - Part of the SONTEL GEANT3 application                      C
C-----------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                          C
C Last modification: 08/16/2002                                         C
C=======================================================================C

#include "geant321/pilot.h"

C=======================================================================C
      SUBROUTINE GUSTEP
C-----------------------------------------------------------------------C
C     User routine called at the end of each tracking step              C
C=======================================================================C

#include "geant321/gcbank.inc"
#include "geant321/gctmed.inc"
#include "geant321/gckine.inc"
#include "geant321/gcking.inc"
#include "geant321/gcflag.inc"
#include "geant321/gctrak.inc"
#include "geant321/gcvolu.inc"
#include "geant321/gccuts.inc"
#include "geant321/gconsp.inc"
#include "geant321/gcscan.inc"
#include "celoss.inc"
#include "pvolum.inc"
#include "sontel.inc"

      IF(SCANFL) THEN
         CALL GSCANU
      ELSE

C --- something generated? ---
         IF(NGKINE.GT.0) THEN
            DO 10 I=1,NGKINE
               CALL GSKING(I)
               IF(ISWIT(9).NE.0)GO TO 10
   10       CONTINUE
         ENDIF
         IF(ISWIT(9).NE.0) GO TO 50

         NVOL=LVOLUM(NLEVEL)

C --- Proportional counters ---
         IF ((NVOL.EQ.IPCG1).OR.(NVOL.EQ.IPCG2)) THEN
          ETOTCOUNTER(NUMBER(2))=ETOTCOUNTER(NUMBER(2))+DESTEP*1000.
          ETOTCOUNTERS=ETOTCOUNTERS+DESTEP*1000.
          IF (ETOTCOUNTER(NUMBER(2)).GT.EIONI) THEN
           IVETO=2
          ENDIF
         ELSE
          IF (NVOL.EQ.IDPCG) THEN
           ETOTCOUNTER(NUMBER(2))=ETOTCOUNTER(NUMBER(2))+DESTEP*1000.
           ETOTCOUNTERS=ETOTCOUNTERS+DESTEP*1000.
          ENDIF
         ENDIF

C --- Scintillators ---
         IF (NVOL.EQ.ISCIN) THEN
          ETOTSCINT(NUMBER(2))=ETOTSCINT(NUMBER(2))+DESTEP*1000.
          ETOTSCINTS=ETOTSCINTS+DESTEP*1000.
         ENDIF
```

```
C --- Energy deposited---

         IF(DESTEP.NE.0.)THEN
          DEDL = DEDL + DESTEP
         ENDIF

C --- debugs event ---

        IF (ISONTELDEBUG.NE.0) THEN
         WRITE(6,3111),NLEVEL,NAMES(NLEVEL),NUMBER(2)
         CALL GPCXYZ
        ENDIF
  50    CALL GDEBUG

 3111 FORMAT(I3,1X,A4,1X,I3)
      ENDIF

      END
```

A.1.8 gutrev.F

```
C=======================================================================C
C GUTREV.F - Part of the SONTEL GEANT3 application                      C
C-----------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                         C
C Last modification: 08/16/2002                                        C
C=======================================================================C

#include "geant321/pilot.h"

C=======================================================================C
      SUBROUTINE GUTREV
C-----------------------------------------------------------------------C
C     User routine to control tracking of one event                     C
C=======================================================================C

      CALL GTREVE

      END
```

A.1.9 ufiles.F

```
C=====================================================================C
C UFILES.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UFILES
C---------------------------------------------------------------------C
C     Routine to handle FFREAD and HBOOK files                        C
C=====================================================================C

      CHARACTER*(*) FILNAM, FSTAT

#if defined(CERNLIB_CRAY)||defined(CERNLIB_UNIX)||defined(CERNLIB_VAX)
      PARAMETER (FILNAM='sontel.ffk')
#endif

#if defined(CERNLIB_IBM)
      PARAMETER (FILNAM='/SONTEL.FFK *')
#endif

#if defined(CERNLIB_CRAY)||defined(CERNLIB_UNIX)
      PARAMETER (FSTAT='OLD')
#endif

#if defined(CERNLIB_VAX)||defined(CERNLIB_IBM)
      PARAMETER (FSTAT='UNKNOWN')
#endif

#if defined(CERNLIB_QFMSOFT)
      OPEN(UNIT=5,FILE=FILNAM,STATUS='OLD',FORM='FORMATTED')
#elif  defined(CERNLIB_VAX)
      OPEN(UNIT=5,FILE=FILNAM,STATUS=FSTAT,READONLY,FORM='FORMATTED')
#elif 1
      OPEN(UNIT=5,FILE=FILNAM,STATUS=FSTAT,FORM='FORMATTED')
#endif

      END
```

A.1.10 ugeom.F

```
C=====================================================================C
C UGEOM.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UGEOM
C---------------------------------------------------------------------C
C     Defines user geometry set up                                    C
C=====================================================================C

#include "geant321/gcbank.inc"
#include "geant321/gckine.inc"
#include "pvolum.inc"
#include "sontel.inc"

      DIMENSION PAR(10)
      DIMENSION AAIR(2), ZAIR(2),WAIR(2)
      DIMENSION APOLY(2), ZPOLY(2),WPOLY(2)
      DIMENSION ACOUNTERGAS(3), ZCOUNTERGAS(3),WCOUNTERGAS(3)
      DIMENSION ASTAINLESS(3), ZSTAINLESS(3),WSTAINLESS(3)

C --- Data ---

      DATA  AAIR/14.01,16./ZAIR/7.,8./WAIR/.7,.3/
      DATA  APOLY/1.01,12.01/ZPOLY/1.,6./WPOLY/8.,8./
      DATA  ACOUNTERGAS/1.01,12.01,39.984/
      DATA  ZCOUNTERGAS/1.,6.,18./
      DATA  WCOUNTERGAS/.0252,.0748,.9/

      DATA  ASTAINLESS/55.845,51.961,58.69/
      DATA  ZSTAINLESS/26.,24.,28./
      DATA  WSTAINLESS/.74,.18,.08/

C --- Defines user particular materials ---

      CALL GSMATE(1,'VACUUM$',1.E-16,1.E-16,1.E-16,1.E+16,1.E+16,0,0)
      CALL GSMIXT(2,'AIR$',AAIR,ZAIR,0.00127,2,WAIR)
      CALL GSMIXT(3,'POLY$',APOLY,ZPOLY,1.032,-2,WPOLY)
      CALL GSMIXT(4,'COUNTERGAS$',ACOUNTERGAS,ZCOUNTERGAS,0.000668,
     $3,WCOUNTERGAS)
      CALL GSMIXT(5,'STAINLESS$',ASTAINLESS,ZSTAINLESS,7.9,
     $3,WSTAINLESS)

C --- Defines user tracking media parameters ---

      IFIELD =  0
      FIELDM =  0.
      TMAXFD =  .1
      DMAXMS =  10.
      DEEMAX =  0.1
      EPSIL  =  0.05
      STMIN  =  1.
      CALL GSTMED( 1,'VACUUM$',1,0,IFIELD,FIELDM,TMAXFD,
     *     SVACDMAXMS,SVACDEEMAX,SVACEPSIL,SVACSTMIN,0,0 )
      CALL GSTMED( 2,'AIR$',2,0,IFIELD,FIELDM,TMAXFD,
     *     DMAXMS,DEEMAX,EPSIL,STMIN,0,0 )
      CALL GSTMED( 3,'POLY$',3,1,IFIELD,FIELDM,TMAXFD,
     *     SPOLYDMAXMS,DEEMAX,EPSIL,STMIN,0,0 )
```

```
      CALL GSTMED( 4,'COUNTERGAS$',4,2,IFIELD,FIELDM,TMAXFD,
     *     DMAXMS,DEEMAX,EPSIL,STMIN,0,0 )
      CALL GSTMED( 6,'COUNTERGAS1$',4,3,IFIELD,FIELDM,TMAXFD,
     *     DMAXMS,DEEMAX,EPSIL,STMIN,0,0 )
      CALL GSTMED( 5,'STAINLESS$',5,0,IFIELD,FIELDM,TMAXFD,
     *     DMAXMS,DEEMAX,EPSIL,STMIN,0,0 )

C --- Redefines tracking parameters ---

      CALL GSTPAR(1,'CUTHAD',1.E-5)
      CALL GSTPAR(2,'CUTHAD',1.E-5)

C --- Rotation matrix ---

      CALL GSROTM(1,90.,0.,0.,90.,90.)
      CALL GSROTM(2,0.,0.,90.,90.,90.,0.)

C === Defines user volumes ===

C --- LABO ---

      PAR(1) = 500.
      PAR(2) = 500.
      PAR(3) = 500.
      CALL GSVOLU('LABO','BOX ',2,PAR,3,IVOL)
      WRITE(6,*) "ugeom",IVOL

C --- General Parameters ---

C     veto counters
      PCounterLength = 200.
      PCounterLength2 = 250.
      PCounterOutside = 5.05
      PCounterInside = 4.875
      NPCounterMater = 5
      NPCounterGas = 4
      NbOfPCounters = 72
      NbOfPCounters2 = 52

C     directional counters
      DPCounterLength = 185.
      DPCounterOutside = 5.05
      DPCounterInside = 4.875
      NDPCounterMater = 5
      NDPCounterGas = 6
      NbOfDPCounters = 40
      NbOfDPCperLayer = 20

C     scintillators
      ScintillatorLength = 100.
      ScintillatorHeight = 40.
      NScintillatorMater = 3
      NbOfScintillators = 4

C --- Definition of carriers ---

      DetectorHeight = 250.
      DetectorWidth = 240.
      DetectorDepth = 220.
      NCarrierMedium = 5
      NAir=2
      CarrierThickness = 0.3
```

```
C --- vertical carrier ---
      PAR(1) = 2.5
      PAR(2) = 2.5
      PAR(3) = 125.
      CALL GSVOLU('VCAR','BOX ',NCarrierMedium,PAR,3,IVOL)
      PAR(1) = 2.5-CarrierThickness
      PAR(2) = 2.5-CarrierThickness
      PAR(3) = 125.-CarrierThickness
      CALL GSVOLU('VCAI','BOX ',NAir,PAR,3,IVOL)
      CALL GSPOS('VCAI',1,'VCAR',0.,0.,0.,0,'ONLY')
      Xpos = DetectorWidth / 2. - 2.5
      Ypos = DetectorDepth / 2. - 2.5
      Zpos = DetectorHeight / 2.
      CALL GSPOS('VCAR',1,'LABO',Xpos,Ypos,Zpos,0,'ONLY')
      CALL GSPOS('VCAR',2,'LABO',Xpos,-Ypos,Zpos,0,'ONLY')
      CALL GSPOS('VCAR',3,'LABO',-Xpos,Ypos,Zpos,0,'ONLY')
      CALL GSPOS('VCAR',4,'LABO',-Xpos,-Ypos,Zpos,0,'ONLY')

C --- horizontal carrier short ---
      PAR(1) = 2.5
      PAR(2) = DetectorDepth / 2. - 5.
      PAR(3) = 5.
      CALL GSVOLU('HCSH','BOX ',NCarrierMedium,PAR,3,IVOL)
      PAR(1) = 2.5-CarrierThickness
      PAR(2) = DetectorDepth / 2. - 5.
      PAR(3) = 5.-CarrierThickness
      CALL GSVOLU('HCSI','BOX ',NAir,PAR,3,IVOL)
      CALL GSPOS('HCSI',1,'HCSH',0.,0.,0.,0,'ONLY')
      Zpos = DetectorHeight - 2. * PCounterOutside - 5.
      CALL GSPOS('HCSH',1,'LABO',Xpos,0,Zpos,0,'ONLY')
      CALL GSPOS('HCSH',2,'LABO',-Xpos,0,Zpos,0,'ONLY')

C --- horizontal carrier long ---
      PAR(1) = DetectorWidth / 2. - 5.
      PAR(2) = 2.5
      PAR(3) = 5.
      CALL GSVOLU('HCLO','BOX ',NCarrierMedium,PAR,3,IVOL)
      PAR(1) = DetectorWidth / 2. - 5.
      PAR(2) = 2.5-CarrierThickness
      PAR(3) = 5.-CarrierThickness
      CALL GSVOLU('HCLI','BOX ',NAir,PAR,3,IVOL)
      CALL GSPOS('HCLI',1,'HCLO',0.,0.,0.,0,'ONLY')
      Zpos = DetectorHeight - 2. * PCounterOutside - 5.
      CALL GSPOS('HCLO',1,'LABO',0,Ypos,Zpos,0,'ONLY')
      CALL GSPOS('HCLO',2,'LABO',0,-Ypos,Zpos,0,'ONLY')

C --- Base carrier ---
      PAR(1) = DetectorWidth / 2. - 5.
      PAR(2) = DetectorDepth / 2. - 5.
      PAR(3) = CarrierThickness / 2.
      CALL GSVOLU('BCAR','BOX ',NCarrierMedium,PAR,3,IVOL)
      PAR(1) =  DetectorWidth / 2. - 15.
      PAR(2) =  90. / 2.
      PAR(3) =  CarrierThickness / 2.
      CALL GSVOLU('BCHO','BOX ',NAir,PAR,3,IVOL)
      CALL GSPOS('BCHO',1,'BCAR',0.,-50.,0.,0,'ONLY')
      CALL GSPOS('BCHO',2,'BCAR',0.,50.,0.,0,'ONLY')
      Xpos = 0.
      Ypos = 0.
      Zpos = 65. - CarrierThickness / 2.
      CALL GSPOS('BCAR',1,'LABO',Xpos,Ypos,Zpos,0,'ONLY')
      Zpos= 65. - (20. - CarrierThickness / 2.)
      CALL GSPOS('BCAR',2,'LABO',Xpos,Ypos,Zpos,0,'ONLY')
```

```
C --- vertical base carrier long ---
      PAR(1) = DetectorWidth / 2. - 5.
      PAR(2) = CarrierThickness / 2.
      PAR(3) =  10.  - CarrierThickness
      CALL GSVOLU('VBCL','BOX ',NCarrierMedium,PAR,3,IVOL)
      Xpos = 0.
      Ypos = DetectorDepth / 2. - 10.
      Zpos = 65.  - 10.
      CALL GSPOS('VBCL',1,'LABO',Xpos,Ypos,Zpos,0,'ONLY')
      Ypos=0.
      CALL GSPOS('VBCL',2,'LABO',Xpos,Ypos,Zpos,0,'ONLY')
      Ypos = -(DetectorDepth / 2. - 10. )
      CALL GSPOS('VBCL',3,'LABO',Xpos,Ypos,Zpos,0,'ONLY')

C --- vertical base carrier short ---
      PAR(1) = CarrierThickness / 2.
      PAR(2) = (DetectorDepth / 2. - 10. - CarrierThickness) / 2.
      PAR(3) =  10.  - CarrierThickness
      CALL GSVOLU('VBCS','BOX ',NCarrierMedium,PAR,3,IVOL)
      Xpos = DetectorWidth / 2. - 10.
      Ypos = (DetectorDepth / 2. - 10.) / 2.
      CALL GSPOS('VBCS',1,'LABO',Xpos,Ypos,Zpos,0,'ONLY')
      CALL GSPOS('VBCS',2,'LABO',Xpos,-Ypos,Zpos,0,'ONLY')
      CALL GSPOS('VBCS',3,'LABO',-Xpos,Ypos,Zpos,0,'ONLY')
      CALL GSPOS('VBCS',4,'LABO',-Xpos,-Ypos,Zpos,0,'ONLY')

C --- Scintillator blocks ---
      CoverBottomLength = 110.
      CoverTopLength = 35.
      CoverPMLength = 30.
      CoverBottomHeight = 43.5
      CoverTopHeight = 70.
      CoverPMHeight = 48.0
      CoverThickness = 0.2
      CoverHeight = CoverBottomHeight + CoverTopHeight + CoverPMHeight
      NCoverMater = 5
      diff = (CoverBottomLength - CoverTopLength) / 2.
      deff = CoverThickness / CoverTopHeight *
     *        SQRT (diff * diff + CoverTopHeight * CoverTopHeight)
      CoverTopGasLength = CoverTopLength + 2. * CoverThickness
     *  * diff / CoverTopHeight

C --- Scintillator cover ---
      PAR(1) = CoverBottomLength / 2.
      PAR(2) = CoverBottomLength /2.
      PAR(3) = CoverHeight / 2.
      CALL GSVOLU('SCCO','BOX ',NAir,PAR,3,IVOL)
      ScintillatorBottom = 65.
      Zpos = ScintillatorBottom + CoverHeight / 2.
      Xpos= .5 * CoverBottomLength
      Ypos = .5 * CoverBottomLength
      CALL GSPOS('SCCO',1,'LABO',Xpos,Ypos,Zpos,0,'ONLY')
      CALL GSPOS('SCCO',2,'LABO',Xpos,-Ypos,Zpos,0,'ONLY')
      CALL GSPOS('SCCO',3,'LABO',-Xpos,Ypos,Zpos,0,'ONLY')
      CALL GSPOS('SCCO',4,'LABO',-Xpos,-Ypos,Zpos,0,'ONLY')

C --- PM cover total ---
      PAR(1) = CoverPMLength / 2.
      PAR(2) =  CoverPMLength /2.
      PAR(3) = CoverPMHeight / 2.
      CALL GSVOLU('PMCO','BOX ',NCoverMater,PAR,3,IVOL)
      Zpos=-CoverHeight / 2. + CoverBottomHeight + CoverTopHeight +
     *         CoverPMHeight / 2.
      CALL GSPOS('PMCO',1,'SCCO',0.,0.,Zpos,0,'ONLY')
```

```
C --- PM cover gas ---
      PAR(1) = CoverPMLength / 2. - CoverThickness
      PAR(2) =  CoverPMLength /2. - CoverThickness
      PAR(3) = CoverPMHeight / 2. - CoverThickness
      CALL GSVOLU('PMCG','BOX ',NAir,PAR,3,IVOL)
      Zpos=-CoverThickness / 2.
      CALL GSPOS('PMCG',1,'PMCO',0.,0.,Zpos,0,'ONLY')

C --- upper cover ---
      PAR(1) = CoverBottomLength / 2.
      PAR(2) = CoverTopLength / 2.
      PAR(3) = CoverBottomLength / 2.
      PAR(4) = CoverTopLength / 2.
      PAR(5) = CoverTopHeight / 2.
      CALL GSVOLU('UPCO','TRD2',NCoverMater,PAR,5,IVOL)
      Zpos=-CoverHeight / 2. + CoverBottomHeight + CoverTopHeight / 2.
      CALL GSPOS('UPCO',1,'SCCO',0.,0.,Zpos,0,'ONLY')

C --- upper cover gas ---
      PAR(1) = CoverBottomLength / 2. - deff
      PAR(2) = CoverTopGasLength / 2.
      PAR(3) = CoverBottomLength / 2. - deff
      PAR(4) = CoverTopGasLength / 2.
      PAR(5) = (CoverTopHeight - CoverThickness) / 2.
      CALL GSVOLU('UPCG','TRD2',NAir,PAR,5,IVOL)
      Zpos=-CoverThickness / 2.
      CALL GSPOS('UPCG',1,'UPCO',0.,0.,Zpos,0,'ONLY')

C --- lower cover ---
      PAR(1) = CoverBottomLength / 2.
      PAR(2) = CoverBottomLength / 2.
      PAR(3) = CoverBottomHeight / 2.
      CALL GSVOLU('LOCO','BOX ',NCoverMater,PAR,3,IVOL)
      Zpos=-CoverHeight / 2. + CoverBottomHeight / 2.
      CALL GSPOS('LOCO',1,'SCCO',0.,0.,Zpos,0,'ONLY')
      PAR(1) = CoverBottomLength / 2. - CoverThickness
      PAR(2) = CoverBottomLength / 2. - CoverThickness
      PAR(3) = CoverBottomHeight / 2. - CoverThickness / 2.
      CALL GSVOLU('LOCG','BOX ',NAir,PAR,3,IVOL)
      Zpos=CoverThickness / 2.
      CALL GSPOS('LOCG',1,'LOCO',0.,0.,Zpos,0,'ONLY')

C --- Scintillator ---
      PAR(1) = ScintillatorLength / 2.
      PAR(2) = ScintillatorLength / 2.
      PAR(3) = ScintillatorHeight / 2.
      CALL GSVOLU('SCIN','BOX ',NScintillatorMater,PAR,3,IVOL)
      ISCIN=IVOL
      Zpos=-(CoverBottomHeight - CoverThickness) / 2. +
     *                               ScintillatorHeight / 2.
      CALL GSPOS('SCIN',1,'LOCG',0.,0.,Zpos,0,'ONLY')

C --- Veto counters ---
      IF (ISWITCHPC(1).NE.0) THEN

      PAR(1) = 0.
      PAR(2) = PCounterOutside
      PAR(3) = PCounterLength / 2.
      XLayerSpacing = 20.
C     short
      CALL GSVOLU('PCO1','TUBE',NPCounterMater,PAR,3,IVOL)
C     long
      PAR(3) = PCounterLength2 / 2.
      CALL GSVOLU('PCO2','TUBE',NPCounterMater,PAR,3,IVOL)
```

```
C     gas inside
      PAR(1) = 0.
      PAR(2) = PCounterInside
      PAR(3) = 0.5 * PCounterLength - (PCounterOutside -
     *      PCounterInside)
C     short
      CALL GSVOLU('PCG1','TUBE',NPCounterGas,PAR,3,IVOL)
      IPCG1=IVOL
      CALL GSPOS('PCG1',1,'PCO1',0.,0.,0.,0,'ONLY')
C     long
      PAR(3) = 0.5 * PCounterLength2 - (PCounterOutside -
     *      PCounterInside)

      CALL GSVOLU('PCG2','TUBE',NPCounterGas,PAR,3,IVOL)
      IPCG2=IVOL
      CALL GSPOS('PCG2',1,'PCO2',0.,0.,0.,0,'ONLY')
      j=1
      NbOfTopCounters = 20
      NbOfUpperSideCounters = 6
      NbOfLowerSideCounters = 10
      CounterSpacing = 10.2

C --- Positioning of veto tubes ---

C     top tubes
      DO i=1,NbOfTopCounters
      Xpos = 0.
      Ypos = -(NbOfTopCounters - 1) / 2. * CounterSpacing
     * + (i-1) * CounterSpacing
      Zpos =DetectorHeight - PCounterOutside
      CALL GSPOS('PCO2',j,'LABO',Xpos,Ypos,Zpos,2,'ONLY')
      j=j+1
      ENDDO

C     upper western and eastern tubes
      DO i=1, NbOfUpperSideCounters
      Xpos = 0.
      Ypos = DetectorDepth / 2. + PCounterOutside
      Zpos =DetectorHeight - 15.  - PCounterOutside
     *        - (i-1) * CounterSpacing
      CALL GSPOS('PCO2',j,'LABO',Xpos,Ypos,Zpos,2,'ONLY')
      j=j+1
      CALL GSPOS('PCO2',j,'LABO',Xpos,-Ypos,Zpos,2,'ONLY')
      j=j+1
      ENDDO

C     lower western and eastern tubes
      DO i=1, NbOfLowerSideCounters
      Xpos = 0.
      Ypos = DetectorDepth / 2. + PCounterOutside
      Zpos = DetectorHeight - 15. - 3. * PCounterOutside
     *      - 7. - (NbOfUpperSideCounters - 1 +(i-1)) * CounterSpacing
      CALL GSPOS('PCO2',j,'LABO',Xpos,Ypos,Zpos,2,'ONLY')
      j=j+1
      CALL GSPOS('PCO2',j,'LABO',Xpos,-Ypos,Zpos,2,'ONLY')
      j=j+1
      ENDDO

C     upper northern and southern tubes
      DO i=1, NbOfUpperSideCounters
      Xpos = DetectorWidth / 2. + PCounterOutside
      Ypos = 0
      Zpos = DetectorHeight - 15.
     *        - PCounterOutside - (i-1) * CounterSpacing
```

```
      CALL GSPOS('PCO1',j,'LABO',Xpos,Ypos,Zpos,1,'ONLY')
      j=j+1
      CALL GSPOS('PCO1',j,'LABO',-Xpos,Ypos,Zpos,1,'ONLY')
      j=j+1
      ENDDO

C     lower northern and southern tubes
      DO i=1, NbOfLowerSideCounters
      Xpos = DetectorWidth / 2. + PCounterOutside
      Ypos = 0
      Zpos = DetectorHeight - 15. - 3. * PCounterOutside
     *    - 7. - (NbOfUpperSideCounters - 1 +
     *        (i-1)) * CounterSpacing
      CALL GSPOS('PCO1',j,'LABO',Xpos,Ypos,Zpos,1,'ONLY')
      j=j+1
      CALL GSPOS('PCO1',j,'LABO',-Xpos,Ypos,Zpos,1,'ONLY')
      j=j+1
      ENDDO

      ENDIF

C --- Directional counters ---
      IF (ISWITCHPC(2).NE.0) THEN

C     metallic outside
      PAR(1) = 0.
      PAR(2) = DPCounterOutside
      PAR(3) = DPCounterLength / 2.
      XLayerSpacing = 20.
      CALL GSVOLU('DPCO','TUBE',NDPCounterMater,PAR,3,IVOL)
      DO i=1,NbOfDPCperLayer
      Xpos = -(NbOfDPCperLayer - 1) / 2. * CounterSpacing +
     *            (i-1) * CounterSpacing
      Ypos = 0.
      Zpos = DPCounterOutside
      CALL GSPOS('DPCO',j,'LABO',Xpos,Ypos,Zpos,1,'ONLY')
      j=j+1
      Zpos = Zpos + XLayerSpacing
      CALL GSPOS('DPCO',j,'LABO',Xpos,Ypos,Zpos,1,'ONLY')
      j=j+1
      ENDDO

C     counter gas
      PAR(1) = 0.
      PAR(2) = DPCounterInside
      PAR(3) = 0.5 * DPCounterLength - (DPCounterOutside -
     *      DPCounterInside)
      CALL GSVOLU('DPCG','TUBE',NDPCounterGas,PAR,3,IVOL)
      IDPCG=IVOL
      CALL GSPOS('DPCG',1,'DPCO',0.,0.,0.,0,'ONLY')
      ENDIF

C --- Close geometry banks (obligatory system routine) ---
      CALL GGCLOS

      END
```

A.1.11 uginit.F

```
C=====================================================================C
C UGINIT.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UGINIT
C---------------------------------------------------------------------C
C     Routine to initialize GEANT/USER program and read data cards    C
C=====================================================================C

#include "geant321/gcunit.inc"
#include "geant321/gclist.inc"
#include "geant321/gckine.inc"
#include "pvolum.inc"
#include "celoss.inc"
#include "sontel.inc"

C --- Opens user files ---

      CALL UFILES

C --- Initializes GEANT ---

      CALL GINIT

      IPARTICLE=14
      CALL FFKEY('PART',IPARTICLE,1,'INTEGER')
      CALL FFKEY('SRCON',ISOURCEON,1,'INTEGER')
      CALL FFKEY('CENTER',SOCENTER,3,'REAL')
      CALL FFKEY('DISTANCE',SODISTANCE,1,'REAL')
      CALL FFKEY('SORAD',SORADIUS,1,'REAL')
      CALL FFKEY('SPHI',PHI,1,'REAL')
      CALL FFKEY('STHETA',THETA,1,'REAL')
      CALL FFKEY('PHI',PHISTART,1,'REAL')
      CALL FFKEY('THETA',THETASTART,1,'REAL')
      CALL FFKEY('POSI',POSSTART,3,'REAL')
      CALL FFKEY('IONI',EIONI,1,'REAL')
      CALL FFKEY('DISCRI',DISC,4,'REAL')
      CALL FFKEY('SONDBG',ISONTELDEBUG,1,'INTEGER')
      CALL FFKEY('SWITCHPC',ISWITCHPC,2,'INTEGER')
      CALL FFKEY('SPECTRUM',ISPECTRUM,1,'INTEGER')
      CALL FFKEY('DETCENT',DETCENTER,3,'REAL')
      CALL FFKEY('DETSIZ',DETSIZE,3,'REAL')
      CALL FFKEY('CROSSDET',ICHECKCROSSING,1,'INTEGER')
      CALL FFKEY('NRTHETA',NRTHETA,1,'INTEGER')
      CALL FFKEY('NRPHI',NRPHI,1,'INTEGER')
      CALL FFKEY('ZENITHINT',ZENITHINT,1,'REAL')
      CALL FFKEY('COSEXP',COSEXP,1,'REAL')
      CALL FFKEY('TIME',TIME,1,'REAL')
      CALL FFKEY('FILENR',IFILENR,4,'INTEGER')
      CALL FFKEY('VDMAXMS',SVACDMAXMS,1,'REAL')
      CALL FFKEY('VDEEMAX',SVACDEEMAX,1,'REAL')
      CALL FFKEY('VEPSIL',SVACEPSIL,1,'REAL')
      CALL FFKEY('VSTMIN',SVACSTMIN,1,'REAL')
      CALL FFKEY('ADMAXMS',SAIRDMAXMS,1,'REAL')
      CALL FFKEY('ADEEMAX',SAIRDEEMAX,1,'REAL')
      CALL FFKEY('AEPSIL',SAIREPSIL,1,'REAL')
      CALL FFKEY('ASTMIN',SAIRSTMIN,1,'REAL')
```

```
      CALL FFKEY('PDMAXMS',SPOLYDMAXMS,1,'REAL')
      CALL FFKEY('PDEEMAX',SPOLYDEEMAX,1,'REAL')
      CALL FFKEY('PEPSIL',SPOLYEPSIL,1,'REAL')
      CALL FFKEY('PSTMIN',SPOLYSTMIN,1,'REAL')
      CALL FFKEY('GDMAXMS',SGASDMAXMS,1,'REAL')
      CALL FFKEY('GDEEMAX',SGASDEEMAX,1,'REAL')
      CALL FFKEY('GEPSIL',SGASEPSIL,1,'REAL')
      CALL FFKEY('GSTMIN',SGASSTMIN,1,'REAL')
      CALL FFKEY('MDMAXMS',SMETDMAXMS,1,'REAL')
      CALL FFKEY('MDEEMAX',SMETDEEMAX,1,'REAL')
      CALL FFKEY('MEPSIL',SMETEPSIL,1,'REAL')
      CALL FFKEY('MSTMIN',SMETSTMIN,1,'REAL')
      CALL FFKEY('RNDM',NRNDM,2,'INTEGER')

      IKINE=1
      PKINE(1)=0.001
      PKINE(2)=0.01

      CALL GFFGO
      CALL GZINIT
      CALL GPART

      PRINT*,IPARTICLE

C --- Prints version number ---

      WRITE(LOUT,10000)

C --- Geometry and materials description ---

      CALL UGEOM
      CALL GLOOK('MATE',LPRIN,NPRIN,IM)
      CALL GLOOK('TMED',LPRIN,NPRIN,IT)
      CALL GLOOK('VOLU',LPRIN,NPRIN,IV)
      IF(IM.NE.0)CALL GPRINT('MATE',0)
      IF(IT.NE.0)CALL GPRINT('TMED',0)
      IF(IV.NE.0)CALL GPRINT('VOLU',0)

C --- Energy loss and cross-sections initializations ---

      CALL FLINIT
      CALL GMORIN
      CALL GPHYSI


C --- Spectrum initialisation ---

       IF (ISPECTRUM.EQ.1) THEN
        CALL ELECTRON_INTSPEC
        IPARTICLE=3
       ELSE
        IF (ISPECTRUM.EQ.2) THEN
         CALL NEUTRON_INTSPEC
         IPARTICLE=13
        ELSE
         IF (ISPECTRUM.EQ.3) THEN
          CALL PROTON_INTSPEC
          IPARTICLE=14
         ELSE
          IF (ISPECTRUM.EQ.4) THEN
           CALL MUON_INTSPEC
           IPARTICLE=6
          ELSE
```

```
           IF (ISPECTRUM.EQ.5) THEN
            CALL GAMMA_INTSPEC
            IPARTICLE=1
           ELSE
            IF (ISPECTRUM.EQ.6) THEN
             CALL NEUTRON2_INTSPEC
             IPARTICLE=13
            ELSE
             ISPECTRUM=0
            ENDIF
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       ENDIF

C --- Position distribution initialization ---

        IF (ISOURCEON.NE.0) THEN
         CALL NEW_SOURCE_ORIENTATION(PHI,THETA)
        ENDIF

C --- Defines user histograms ---

      CALL UHINIT

C --- Copy number initialization ---

      NCOPY=0

10000 FORMAT(/,'  VERSION 1.00 ',/)

      END
```

A.1.12 uglast.F

```
C=====================================================================C
C UGLAST.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UGLAST
C---------------------------------------------------------------------C
C     Termination routine to print histograms and statistics          C
C=====================================================================C

#include "geant321/gcflag.inc"
#include "pvolum.inc"
#include "celoss.inc"

      WRITE(6,*) "uglast"
      CALL GLAST
      CALL IGEND

      END
```

A.1.13 uhinit.F

```
C======================================================================C
C UHINIT.F - Part of the SONTEL GEANT3 application                     C
C----------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                         C
C Last modification: 08/16/2002                                        C
C======================================================================C

#include "geant321/pilot.h"

C======================================================================C
      SUBROUTINE UHINIT
C----------------------------------------------------------------------C
C     Routine to book the user histograms                              C
C======================================================================C

#include "geant321/gckine.inc"
#include "pvolum.inc"

C --- Histograms for shower development ---

      NBINS=BINLIMS(1)

C --- Incident energies ---

      CALL HBOOK1(1,'INCIDENT KINETIC ENERGY$',
     *  1000,40.,20000.,0.)

C --- Start Positions ---

      CALL HBOOK2(2,'START POSITION XY',250,-500,500,250,-500,500,0.)
      CALL HBOOK2(3,'START POSITION YZ',250,-500,500,250,-500,500,0.)
      CALL HBOOK2(4,'START POSITION ZX',250,-500,500,250,-500,500,0.)

C --- Incident direction ---

      CALL HBOOK2(5,'INCIDENT DIRECTION$',181,-1.,361.,202,-1.01,1.01,0.)


C --- Proportional Counter Histograms ---

      CALL HBOOK1(6,'PCOUNTER VETO SIGNAL$',2,-0.5,1.5,0.)
      CALL HBOOK1(7,'ESTIMATED DIRECTION WITH VETO$',5,-2.5,2.5,0.)
      CALL HBOOK1(8,'ESTIMATED DIRECTION WITHOUT VETO$',5,-2.5,2.5,0.)
      CALL HBOOK2(12,'INCIDENT ENERGY OF DETECTED WV$',
     *  1000,40.,20000.,5,0.5,5.5,0.)
      CALL HBOOK2(13,'INCIDENT ENERGY OF DETECTED WOV$',
     *  1000,40.,20000.,5,0.5,5.5,0.)
      CALL HBOOK2(15,'INCIDENT ENERGY OF SCINTILLATING PARTICLE WV$',
     *  1000,40.,20000.,4,0.5,4.5,0.)
      CALL HBOOK2(16,'INCIDENT ENERGY OF SCINTILLATING PARTICLE WOV$',
     *  1000,40.,20000.,4,0.5,4.5,0.)
      CALL HBOOK1(14,'NUMBER OF DETECTED DIRECTIONS',
     *  5,0.5,5.5,0.)

C --- Scintillator Histogram ---

      CALL HBOOK1(9,'SCINTILLATOR DEPOSIT ENERGY$',2000,0.,2000.,0.)
      CALL HBOOK1(10,'SCINTILLATOR CHANNEL WITH VETO$', 4,.5,4.5,0.)
      CALL HBOOK1(11,'SCINTILLATOR CHANNEL WITHOUT VETO$',
     *  4,.5,4.5,0.)

      END
```

A.1.14 celoss.inc

```
#if !defined(CERNLIB_INCLUDE)
      COMMON/CELOSS/ EKIN,DEDL
#endif

#if defined(CERNLIB_INCLUDE)
#include "incbeg.inc"
     +   celoss
#include "incend.inc"
#endif
```

A.1.15 pvoulm.inc

```
#if !defined(CERNLIB_INCLUDE)
      DIMENSION BINLIMS(3)
      COMMON/PVOLUM/ THICK,BINLIMS,IMAT,IPARTICLE
#endif

#if defined(CERNLIB_INCLUDE)
#include "incbeg.inc"
     +   pvolum
#include "incend.inc"
#endif
```

A.1.16 sontel.inc

```
      COMMON/ELECTRONSPEC/ ELEEKIN(50),ELESPECDIFF(50)
     *  ,ELESPECINT(50), NELE
      COMMON/NEUTRONSPEC/ NEUEKIN(50),NEUSPECDIFF(50)
     *   ,NEUSPECINT(50), NNEU
      COMMON/NEUTRONSPEC2/ NEUEKIN2(50),NEUSPECDIFF2(50)
     *   ,NEUSPECINT2(50), NNEU2
      COMMON/MUONSPEC/ MUOEKIN(50),MUOSPECDIFF(50)
     *   ,MUOSPECINT(50), NMUO
      COMMON/PROTONSPEC/ PROEKIN(50),PROSPECDIFF(50)
     *   ,PROSPECINT(50), NPRO
      COMMON/GAMMASPEC/ GAMEKIN(50),GAMSPECDIFF(50)
     *   ,GAMSPECINT(50), NGAM
       COMMON/SELECTSPEC/ SELEKIN(50),SELSPECDIFF(50)
     *   ,SELSPECINT(50), NSEL
      COMMON/SPECHOICE/ISPECTRUM
      COMMON/POSINIT/ POSSTART(3)
      COMMON/DIRINIT/ PHISTART, THETASTART
      COMMON/COPYNR/ NCOPY
      COMMON/DIRPC/ EKININC
      COMMON/PCVETODIR/ ETOTCOUNTER(124), ETOTCOUNTERS,IPCG1,
     * IPCG2,IDPCG, IVETO,IDIR(5), EIONI
      COMMON/SCINTILLATOR/ ETOTSCINT(4), ETOTSCINTS, ICH1, ICH2,
     *  ICH3, ICH4, ISCIN, INIT, DISC(4)
      COMMON/SONDEBUG/ ISONTELDEBUG
      COMMON/SWITCHPC/ ISWITCHPC(2)
      COMMON/SOURCE/SDIRR(3),SDIRTH(3),SDIRPH(3),SOPHI,SOTHETA,
     *      SOCENTER(3),SODISTANCE,SORADIUS,ISOURCEON
      COMMON/DETECTORBOX/DETCENTER(3),DETSIZE(3),ICHECKCROSSING
      COMMON/AIRSTEP/SAIRDMAXMS,SAIRDEEMAX,SAIREPSIL,SAIRSTMIN
      COMMON/POLYSTEP/SPOLYDMAXMS,SPOLYDEEMAX,SPOLYEPSIL,
     *      SPOLYSTMIN
      COMMON/VACSTEP/SVACDMAXMS,SVACDEEMAX,SVACEPSIL,SVACSTMIN
      COMMON/GASSTEP/SGASDMAXMS,SGASDEEMAX,SGASEPSIL,SGASSTMIN
      COMMON/METSTEP/SMETDMAXMS,SMETDEEMAX,SMETEPSIL,SMETSTMIN
      COMMON/MAIN/NRTHETA,NRPHI,ZENITHINT,COSEXP,TIME,IFILENR
```

A.2 Application SOLSPEC

A.2.1 main.F

```
C=====================================================================C
C MAIN.F - Part of the SOLSPEC GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                       C
C Last modification: 08/26/2002                                      C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      PROGRAM SOLSPECB
C---------------------------------------------------------------------C
C     Simulation of atmospheric cascade                              C
C=====================================================================C

#include "geant321/gcflag.inc"
#include "sontel.inc"

      COMMON/PAWC/H(10000000)
      COMMON/GCBANK/Q(50000000)

      REAL ZENITHINT
      REAL PI
      PARAMETER(PI=3.1415926535)
      CHARACTER FILE*4, RUNNAME*5, FILEEXT*6, DIR*6

      CALL TIMEST(1E5)
      CALL GZEBRA(5000000)
      CALL HLIMIT(-1000000)
      CALL UGINIT

C --- Particle beam ---

      INTENSITY=INT(ZENITHINT*TIME)

      DO I=1,INTENSITY
        CALL GTRIGI
        CALL GTRIG
        CALL GTRIGC
      ENDDO

C --- Histogram output ---

      WRITE(RUNNAME,300)IFILENR
300   FORMAT(I5)
      DIR='hbook/'
      FILEEXT='.hbook'
      CALL HRPUT(0,DIR//RUNNAME//FILEEXT,'T')

      CALL UGLAST

      END
```

A.2.2 atmos.F

```
C=====================================================================C
C ATMOS.F - Part of the SOLSPEC GEANT3 application                     C
C---------------------------------------------------------------------C
C Modification of the UCAR stdatm76.F source code available at         C
C  http://dss.ucar.edu/libraries/meteorology/                         C
C Last modification: 08/22/2002 by M. R. Moser, University of Bern, CH C
C=====================================================================C
C
      SUBROUTINE STDZ2P(P,H1,L,TCENT,RHO)
C
C COMPUTES PRESSURE, TEMPERATURE, AND DENSITY VALUES FROM INPUT HEIGHTS
C  BASED ON US STANDARD ATMOSPHERE, 1976
C  VALUES NOT VALID ABOVE 84852 KM.
C
C INPUT
C       H1    - HEIGHT IN FEET OR METERS
C       L     - UNITS FLAG - 0=METERS, 1=FEET
C OUTPUT
C       P     - PRESSURE IN MB
C       TCENT - TEMPERATURE IN DEG C
C       RHO   - DENSITY IN KG/M3
C
      DIMENSION HBASE(10),HTOP(10),TB(10),GRAD(10),ABS(10)
      DIMENSION PB(11),RB(10)
      DATA HBASE/ 0.,1.1E4,2.E4,3.2E4,4.7E4,5.1E4,7.1E4,3*84852.0/
      DATA HTOP /1.1E4,2.E4,3.2E4,4.7E4,5.1E4,7.1E4,4*84852.0/
      DATA TB / 15.,2*-56.5,-44.5,2*-2.5,-58.5,3*-86.2/
      DATA GRAD/-.0065,0.,.001,.0028,0.,-.0028,-.002,3*0./
      DATA CHECK/0./
C     H1 IS ALTITUDE, IF IN METERS SET L .LE. 0.  IF IN FEET SET L .GE. 1
C     P IS PRESSURE IN MB.  TCENT IS TEMP CELSIUS.  RHO IS DENSITY IN KG/M**3
C
      IF(CHECK.NE.0.) GO TO 16
      ABSZ = 273.15
      PZERO=1013.250
      G=980.665
      R = 83143200./28.9644
      PB(1) = PZERO
      DO 3  I=1,8
      ABS(I) = TB(I) + ABSZ
    3 CONTINUE
       DO 15  I=1,8
      IF (GRAD(I) .EQ. 0)  GO TO 10
C  HERE FOR TEMP GRADIENT NOT ZERO
    5 PB(I+1) = PB(I)*((ABS(I+1)/ABS(I))**(100.*G/(-GRAD(I)*R)))
      GO TO 12
C  HERE FOR TEMP GRADIENT ZERO
   10 PB(I+1) = PB(I)*2.7182818**((-100.*G/(R*ABS(I)))*(HTOP(I)-HBASE(I)
     X  ))
   12 RB(I) = 1000.*PB(I)/(R*ABS(I)) * 1000.
   15 CONTINUE
      CHECK=10.
   16 CONTINUE
      IF(L. LT.1) GO TO 30
C
      H = H1*.3048
      GO TO 4
30    H = H1
4     CONTINUE
C  FIND THE ATMOSPHERIC LAYER WE ARE IN
   35 DO 55  I = 1,8
      IF  (H .GT. HBASE(I+1)) GO TO 55
   37 IF (GRAD(I)) 39,43,39
```

```
C  TEMP GRADIENT NOT ZERO
   39 P= PB(I)*((H-HBASE(I))*GRAD(I)/ABS(I)+1)**(-100.*G/(GRAD(I)*R))
      GO TO 45
C  ISOTHERMAL LAYER
   43 P= PB(I)*EXP((H-HBASE(I))*(-G)/(.01*R*ABS(I)))
   45 T = ABS(I) + GRAD(I)*(H-HBASE(I))
      TCENT = T - ABSZ
      RHO = 1.E6* P/(R*T)
      RETURN
   55 CONTINUE
   56 WRITE(6,98)H
98    FORMAT(' STDATM -  ALTITUDE OUT OF RANGE ',F12.2,//)
      END
      SUBROUTINE STDP2Z(P,H,HFT,TCENT,DENS)
C
C COMPUTES HEIGHT, TEMPERATURE, AND DENSITY VALUES FROM INPUT PRESSURES
C  BASED ON US STANDARD ATMOSPHERE, 1976
C  VALUES NOT VALID ABOVE 84852 KM.
C
C INPUT
C       P     - PRESSURE IN MB
C OUTPUT
C       H     - HEIGHT IN METERS
C       HFT   - HEIGHT IN FEET
C       TCENT - TEMPERATURE IN DEG C
C       DENS  - DENSITY IN KG/M3
C
      DIMENSION HBASE(10),HTOP(10),TB(10),GRAD(10),ABS(10),PB(10),RB(10)
      DATA HBASE/ 0.,1.1E4,2.E4,3.2E4,4.7E4,5.1E4,7.1E4,3*84852.0/
      DATA HTOP /1.1E4,2.E4,3.2E4,4.7E4,5.1E4,7.1E4,4*84852.0/
      DATA GRAD/-.0065,0.,.001,.0028,0.,-.0028,-.002,3*0./
      DATA ABS/288.15,216.65,216.65,228.65,270.65,270.65,
     2  214.65,186.95,2*0./
      DATA PB/1013.25,0.22632064E+03,0.54748887E+02,0.86801870E+01,
     2 0.11090631E+01,0.66938874E+00,0.39564205E-01,0.37338360E-02,
     3 2*0./
      ABSZ = 273.15
      G=980.665
      R = 83143200./28.9644
      IF(P  .GT. PB(1)) GO TO 56
   35 DO 55  I = 1,8
      IF (P .GE. PB(I+1))  GO TO 37
   55 CONTINUE
   56 PRINT 1001,P
 1001 FORMAT (' STDATM- PRESSURE OUT OF EXCEEDED ',F8.2)
   37 IF (GRAD(I)) 39,43,39
C  TEMP GRADIENT NOT ZERO
   39 H = ABS(I)/(-GRAD(I)) *(1. -(P/PB(I))**(-GRAD(I)*R/(100.*G)))
     X  + HBASE(I)
      GO TO 45
C  ISOTHERMAL LAYER
   43 H = HBASE(I) + .01*R*ABS(I)/G *(ALOG(PB(I)/P))
   45 T = ABS(I) + GRAD(I)*(H-HBASE(I))
      TCENT = T - ABSZ
      DENS = 1000*1000.*P/(R*T)
      HFT = H/.3048
      RETURN
      END
```

A.2.3 gukine.F

```
C=====================================================================C
C GUKINE.F - Part of the SOLSPEC GEANT3 application                   C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/26/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE GUKINE
C---------------------------------------------------------------------C
C     Generates Kinematics for primary track                          C
C=====================================================================C

#include "geant321/gcbank.inc"
#include "geant321/gcflag.inc"
#include "geant321/gckine.inc"
#include "geant321/gconsp.inc"
#include "geant321/gcscan.inc"
#include "celoss.inc"
#include "pvolum.inc"
#include "sontel.inc"

      CHARACTER*20 CHNPAR
      DIMENSION VERTEX(3),PLAB(3),RNDM(1),UB(10)
      DIMENSION NID1(1000),NID2(10000),NID3(10000)
      DIMENSION NID4(10000),NID5(36000),SDIR(3)
      SAVE NID1, NID2,NID3, NID4,NID5
      DATA NID1/1000*0/NID2/10000*0/NID3/10000*0/NID4/10000*0/
      DATA NID5/36000*0/
      SAVE VERTEX,PLAB
      DATA VERTEX/0.,1.,70./
      DATA PLAB  /3*0./
      COMMON/PRODEVENTS/NPRODEVENTS

      IF(SCANFL) THEN
         CALL GSCANK
      ELSE
         DEDL=0.
         IF (ISPECTRUM.NE.0) THEN
          EKIN=COMPUTE_ENERGY(SELEKIN,SELSPECDIFF,SELSPECINT, NSEL)
         ELSE
          IF(IKINE.EQ.0) THEN
           EKIN=PKINE(1)
          ELSE
           CALL GRNDM(RNDM,1)
           EKIN=PKINE(1)+(PKINE(2)-PKINE(1))*RNDM(1)
          ENDIF
         ENDIF
         CALL GFPART(IPARTICLE,CHNPAR,ITRTYP,AMASS,CHARGE,TLIFE,UB,NWB)
         ETOT=EKIN+AMASS
         PLAB(3)=SQRT((ETOT+AMASS)*(ETOT-AMASS))
         VERTEX(1)=POSSTART(1)
         VERTEX(2)=POSSTART(2)
         VERTEX(3)=POSSTART(3)
         fac=3.1415926/180.
         THETA=THETASTART*fac
         PHI=PHISTART*fac
         SDIR(1)=SIN(theta)*COS(phi)
         SDIR(2)=SIN(theta)*SIN(phi)
         SDIR(3)=COS(theta)
```

```
         PLAB(1)=PLAB(3)*SDIR(1)
         PLAB(2)=PLAB(3)*SDIR(2)
         PLAB(3)=PLAB(3)*SDIR(3)
         NPRODEVENTS=NPRODEVENTS+1
         CALL GSVERT(VERTEX,0,0,0,0,NVERT)
         CALL GSKINE(PLAB,IPARTICLE,NVERT,0,0,NT)

C --- register primary particle ---
         EKININC=EKIN*1000.
         CALL HFF1(1,NID1,EKIN*1000.,1.)
         CALL HFF2(2,NID2,VERTEX(1),VERTEX(2),1.)
         CALL HFF2(3,NID3,VERTEX(2),VERTEX(3),1.)
         CALL HFF2(4,NID4,VERTEX(3),VERTEX(1),1.)
         CALL HFF2(5,NID5,phi/fac,COS(theta),1.)

C --- kinematics debug (controlled by ISWIT(1)) ---
         IF(IDEBUG.EQ.1) THEN
            IF(ISWIT(1).EQ.1) THEN
               CALL GPRINT('VERT',0)
               CALL GPRINT('KINE',0)
            ENDIF
         ENDIF
      ENDIF
      END


C=====================================================================C
      SUBROUTINE PROTON_INTSPEC
C---------------------------------------------------------------------C
C     Normalizes integral and differential proton spectrum            C
C=====================================================================C

#include "sontel.inc"

      DIMENSION  DATAEKIN(18), DATASPEC(18)

      DATA DATAEKIN/3500.,4500.,5500.,6500.,7500.,8500.,9500.,10500.,
     *      11500.,12500.,13500.,14500.,15500.,16500.,17500.,18500.,
     *      19500.,20500./

      DATA DATASPEC/41.8341,10.7611,3.5371,1.3761,0.6061,0.2931,0.1531,
     *       0.0851,0.0491,0.0301,0.0191,0.0121,0.0082031,0.0056141,
     *       0.0039241,0.0027961,0.0020271,0.0014921/

      EMIN=4000.
      EMAX=20000.
      NPRO=18
      NSEL=18
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,PROEKIN,PROSPECDIFF,
     *                     PROSPECINT,NPRO, EMIN,EMAX)
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,SELEKIN,SELSPECDIFF,
     *                     SELSPECINT,NSEL, EMIN,EMAX)
      END


C=====================================================================C
      FUNCTION COMPUTE_ENERGY(EK,SPECDIFF,SPECINT,NPOINT)
C---------------------------------------------------------------------C
C     Computes initial energy                                         C
C=====================================================================C

      DIMENSION RNDM(1),EK(50), SPECDIFF(50), SPECINT(50)
```

```
      CALL GRNDM(RNDM,1)
      DO i=2, NPOINT
        IF ((RNDM(1).GE.SPECINT(i-1))
     *                  .AND.(RNDM(1).LE.SPECINT(i))) THEN
        x0=EK(i-1)
        x1=EK(i)
        A0=SPECDIFF(i-1)
        A1=SPECDIFF(i)
        REST=RNDM(1)-SPECINT(i-1)
        Ax=rest*ALOG(A1/A0)/(x1-x0) + A0
        EKIN=((x1-x0)*ALOG(Ax/A0)/ALOG(A1/A0)) + x0
        COMPUTE_ENERGY=EKIN/1000.
        RETURN
        ENDIF
      ENDDO
      END


C=====================================================================C
      SUBROUTINE COMPUTE_INTSPEC(DATAEK,DATASPEC,EK,SPECDIFF,
     *                                   SPECINT,NPOINT, EMIN,EMAX)
C---------------------------------------------------------------------C
C     Computes integral spectrum                                      C
C=====================================================================C

#include "sontel.inc"

      DIMENSION  DATAEK(50), DATASPEC(50), EK(50), SPECDIFF(50)
      DIMENSION  SPECINT(50)

      EK(1)=EMIN
      x=EMIN
      x0=DATAEK(1)
      x1=DATAEK(2)
      A0=DATASPEC(1)
      A1=DATASPEC(2)
      SPECDIFF(1)=A0*( (A1/A0) ** ((x-x0)/(x1-x0)) )
      EK(NPOINT)=EMAX
      x=EMAX
      x0=DATAEK(NPOINT-1)
      x1=DATAEK(NPOINT)
      A0=DATASPEC(NPOINT-1)
      A1=DATASPEC(NPOINT)
      SPECDIFF(NPOINT)=A0*( (A1/A0) ** ((x-x0)/(x1-x0)) )
      DO i=2, NPOINT-1
        SPECDIFF(i)=DATASPEC(i)
        EK(i)=DATAEK(i)
      ENDDO
      SPECINT(1)=0.
      DO i=2, NPOINT
        x0=EK(i-1)
        x1=EK(i)
        A0=SPECDIFF(i-1)
        A1=SPECDIFF(i)
        SPECINT(i)=SPECINT(i-1)+ ((x1-x0)*(A1-A0)/ALOG(A1/A0))
      ENDDO
      DO i=1, NPOINT
       SPECDIFF(i)=SPECDIFF(i)/SPECINT(NPOINT)
       SPECINT(i)=SPECINT(i)/SPECINT(NPOINT)
      ENDDO
      END
```

A.2.4 guout.F

```
C=======================================================================C
C GUOUT.F - Part of the SOLSPEC GEANT3 application                      C
C-----------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                          C
C Last modification: 08/22/2002                                         C
C=======================================================================C

#include "geant321/pilot.h"

C=======================================================================C
      SUBROUTINE GUOUT
C-----------------------------------------------------------------------C
C     User routine called at the end of each event                      C
C=======================================================================C

      END
```

A.2.5 gustep.F

```
C=====================================================================C
C GUSTEP.F - Part of the SOLSPEC GEANT3 application                    C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/26/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE GUSTEP
C---------------------------------------------------------------------C
C     User routine called at the end of each tracking step            C
C=====================================================================C

#include "geant321/gcbank.inc"
#include "geant321/gctmed.inc"
#include "geant321/gckine.inc"
#include "geant321/gcking.inc"
#include "geant321/gcflag.inc"
#include "geant321/gctrak.inc"
#include "geant321/gcvolu.inc"
#include "geant321/gccuts.inc"
#include "geant321/gconsp.inc"
#include "geant321/gcscan.inc"
#include "celoss.inc"
#include "pvolum.inc"
#include "sontel.inc"

      DIMENSION NID(1000)
      DIMENSION NID6(50)
      DIMENSION IPARTBIN(20)
      DATA IPARTBIN/1,2,3,0,4,5,0,6,7,0,0,0,8,9,9,0,0,0,0,0/

      IF (SCANFL) THEN
         CALL GSCANU
      ELSE

C --- something generated? ---

         IF (NGKINE.GT.0) THEN
            DO 10 I=1,NGKINE
               CALL GSKING(I)
               IF(ISWIT(9).NE.0)GO TO 10
   10       CONTINUE
         ENDIF
         IF(ISWIT(9).NE.0) GO TO 50

C --- registers and stops particles in the detection layer ---

         NVOL=LVOLUM(NLEVEL)
         IF ((NVOL.EQ.IDETL).AND.(GEKIN.GT.0)) THEN
           EKIN = GEKIN*1000.
           EX = VECT(4)
           EY = VECT(5)
           EZ = VECT(6)
           IDPART = IPART
           ECOST=-EZ
           EPHI=ATAN2(-EY,-EX)
           ITHIST=0
           DO I=0,NRTHETA-1
             THMIN=1.-I/(NRTHETA-1.)-0.5/(NRTHETA-1.)
             THMAX=1.-I/(NRTHETA-1.)+0.5/(NRTHETA-1.)
```

```
            IF ((ECOST.GT.THMIN).AND.(ECOST.LE.THMAX))
     *       ITHIST=I
           ENDDO
           IPHIST=0
           NRPHIMULT=4*(NRPHI-1)+1
           DO I=0,NRPHIMULT-1
            PHMIN=-pi-pi/(4.*(NRPHI-1.))+I*pi/(2.*(NRPHI-1.))
            PHMAX=-pi+pi/(4.*(NRPHI-1.))+I*pi/(2.*(NRPHI-1.))
            IF ((EPHI.GT.PHMIN).AND.(EPHI.LE.PHMAX)) IPHIST=I
           ENDDO

           IF (IDPART.LE.20) THEN
             IF (IPARTBIN(IDPART).NE.0) THEN
              JHISTO=IPARTBIN(IDPART)*1000+ITHIST*100+IPHIST
              CALL HFF1(JHISTO,NID,EKIN,1.)
              JHISTO=IPARTBIN(IDPART)+10
              CALL HFF1(JHISTO,NID,EKIN,1.)
             ENDIF
           ENDIF
           CALL HFF1(6,NID6,1.*IDPART,1.)
           GEKIN=0.
           VECT(7)=0.
         ENDIF

C --- debugs event ---

        IF (ISONTELDEBUG.NE.0) THEN
         WRITE(6,3111),NLEVEL,NAMES(NLEVEL),NUMBER(2)
         CALL GPCXYZ
        ENDIF
  50    CALL GDEBUG

 3111 FORMAT(I3,1X,A4,1X,I3)
      ENDIF

      END
```

A.2.6 ugeom.F

```
C=====================================================================C
C UGEOM.F - Part of the SONTEL GEANT3 application                     C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/16/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UGEOM
C---------------------------------------------------------------------C
C     Defines user geometry set up                                    C
C=====================================================================C

#include "geant321/gcbank.inc"
#include "geant321/gckine.inc"
#include "pvolum.inc"
#include "sontel.inc"

      DIMENSION PAR(10)
      DIMENSION AAIR(2), ZAIR(2),WAIR(2)
      DIMENSION PRESS(101)
      DIMENSION HEIGHT(101)
      DIMENSION DENSITY(101)
      CHARACTER XNR*4
      INTEGER LAY
      REAL DPRESS

C --- Data ---

      DATA LAY/100/
      DATA AAIR/14.01,16./ZAIR/7.,8./WAIR/.762,.238/

C --- Defines user tracking media ---

      IFIELD =  0
      FIELDM =  0.
      TMAXFD =  .1
      DMAXMS =  10000.
      DEEMAX =  0.1
      EPSIL  =  0.05
      STMIN  =  1.
      PAR(1) = 1.E7
      PAR(2) = 1.E7
      PAR(3) = 1.E7

      CALL GSMATE(300,'VACUUM$',1.E-16,1.E-16,1.E-16,1.E+16,1.E+16,0,0)
      CALL GSTMED(300,'VACUUM$',300,0,IFIELD,FIELDM,TMAXFD,
     *     DMAXMS,DEEMAX,EPSIL,STMIN,0,0 )
      CALL GSVOLU('LABO','BOX ',300,PAR,3,IVOL)

C --- Defines atmospheric layers ---

      PRESSMIN=50.
      CALL STDZ2P(PRESSGG,3135.,0,TEMP,DENS)
      DPRESS=(PRESSGG-PRESSMIN)/LAY
      DO I=1,LAY+1
         PRESS(I)=PRESSGG-(I-1.)*DPRESS
         CALL STDP2Z(PRESS(I),HEIGHT(I),
     *        HFT,TEMP,DENSITY(I))
      ENDDO
```

```
      DO I=1,LAY
         WRITE(XNR,100)I+100
100      FORMAT('A',I3)
         PRESS1=PRESS(I)
         RHOI=0.
         DP=(PRESS(I+1)-PRESS(I))/1000.
300      CALL STDP2Z(PRESS1,H1,HFT1,T1,D1)
         CALL STDP2Z(PRESS1+DP,H2,HFT2,T2,D2)
         DH=H2-H1
         RHO=(D1+D2)/2.
         RHOI=RHOI+RHO*DH
         PRESS1=PRESS1+DP
         IF (PRESS1.GT.PRESS(I+1)) GOTO 300
         RMEAN=RHOI/(HEIGHT(I+1)-HEIGHT(I))/1000.
         DMAXMS=(HEIGHT(I+1)-HEIGHT(I))*LAY*1.E0
         CALL GSMIXT(I,XNR,AAIR,ZAIR,RMEAN,2,WAIR)
         CALL GSTMED(I,XNR,I,0,IFIELD,FIELDM,TMAXFD,
     *        DMAXMS,DEEMAX,EPSIL,STMIN,0,0 )
         CALL GSTPAR(I,'CUTHAD',1.E-5)
         PAR(1) = 1.E7
         PAR(2) = 1.E7
         PAR(3) = 0.5E2*(HEIGHT(I+1)-HEIGHT(I))
         CALL GSVOLU(XNR,'BOX ',I,PAR,3,IVOL)
         CALL GSPOS(XNR,1,'LABO',0.,0.,
     *        0.5*1.E2*(HEIGHT(I+1)+HEIGHT(I)),0,'ONLY')
      ENDDO

      PAR(1) = 1.E7
      PAR(2) = 1.E7
      PAR(3) = 0.5E2*HEIGHT(1)
      CALL GSTMED(301,'DETL',300,0,IFIELD,FIELDM,TMAXFD,
     *        DMAXMS,DEEMAX,EPSIL,STMIN,0,0 )
      CALL GSVOLU('DETL','BOX ',301,PAR,3,IVOL)
      IDETL=IVOL
      CALL GSPOS('DETL',1,'LABO',0.,0.,0.5E2*HEIGHT(1),0,'ONLY')
      CALL GGCLOS

      END
```

A.2.7 uginit.F

```
C=====================================================================C
C UGINIT.F - Part of the SOLSPEC GEANT3 application                   C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/26/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UGINIT
C---------------------------------------------------------------------C
C     Routine to initialize GEANT/USER program and read data cards    C
C=====================================================================C

#include "geant321/gcunit.inc"
#include "geant321/gclist.inc"
#include "geant321/gckine.inc"
#include "geant321/gcflag.inc"
#include "pvolum.inc"
#include "celoss.inc"
#include "sontel.inc"

C --- Opens user files ---

      CALL UFILES

C --- Initializes GEANT ---

      CALL GINIT

      IPARTICLE=14
      CALL FFKEY('PART',IPARTICLE,1,'INTEGER')
      CALL FFKEY('SRCON',ISOURCEON,1,'INTEGER')
      CALL FFKEY('PHI',PHISTART,1,'REAL')
      CALL FFKEY('THETA',THETASTART,1,'REAL')
      CALL FFKEY('POSI',POSSTART,3,'REAL')
      CALL FFKEY('SONDBG',ISONTELDEBUG,1,'INTEGER')
      CALL FFKEY('SPECTRUM',ISPECTRUM,1,'INTEGER')
      CALL FFKEY('NRTHETA',NRTHETA,1,'INTEGER')
      CALL FFKEY('NRPHI',NRPHI,1,'INTEGER')
      CALL FFKEY('ZENITHINT',ZENITHINT,1,'REAL')
      CALL FFKEY('COSEXP',COSEXP,1,'REAL')
      CALL FFKEY('TIME',TIME,1,'REAL')
      CALL FFKEY('FILENR',IFILENR,4,'INTEGER')
      CALL FFKEY('RNDM',NRNDM,2,'INTEGER')

      IKINE=1
      PKINE(1)=0.001
      PKINE(2)=0.01

      CALL GFFGO
      CALL GZINIT
      CALL GPART

      PRINT*,IPARTICLE

C --- Prints version number ---

      WRITE(LOUT,10000)
```

```
C --- Geometry and materials description ---

      CALL UGEOM
      CALL GLOOK('MATE',LPRIN,NPRIN,IM)
      CALL GLOOK('TMED',LPRIN,NPRIN,IT)
      CALL GLOOK('VOLU',LPRIN,NPRIN,IV)
      IF(IM.NE.0)CALL GPRINT('MATE',0)
      IF(IT.NE.0)CALL GPRINT('TMED',0)
      IF(IV.NE.0)CALL GPRINT('VOLU',0)

C --- Energy loss and cross-sections initializations ---

      CALL FLINIT
      CALL GMORIN
      CALL GPHYSI

C --- Spectrum initialisation ---

      IF (ISPECTRUM.EQ.3) THEN
       CALL PROTON_INTSPEC
       IPARTICLE=14
      ELSE
       ISPECTRUM=0
      ENDIF

C --- Defines user histograms ---

      CALL UHINIT

C --- Copy number initialization ---

      NCOPY=0

10000 FORMAT(/,'  VERSION 1.00 ',/)

      END
```

A.2.8 uhinit.F

```
C=====================================================================C
C UHINIT.F - Part of the SOLSPEC GEANT3 application                   C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/26/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UHINIT
C---------------------------------------------------------------------C
C     Routine to book the user histograms                             C
C=====================================================================C

#include "geant321/gckine.inc"
#include "pvolum.inc"
#include "sontel.inc"

C --- Histograms for shower development ---

      NBINS=BINLIMS(1)

C --- Directional energy distribution ---

      NRPHIMULT=4*(NRPHI-1)+1
      NRPART=9
      DO I=0,NRTHETA-1
        DO J=0,NRPHIMULT-1
          DO K=1,NRPART
            CALL HBOOK1(1000*K+100*I+J,'ENERGY DISTRIBUTION$',
     *        200,0.,2000.,0.)
          ENDDO
        ENDDO
      ENDDO

C --- Omnidirectional energy distribution ---

      DO K=1,NRPART
        CALL HBOOK1(10+K,'OMNIDIRECTIONAL ENERGY DISTRIBUTION$',
     *    200,0.,2000.,0.)
      ENDDO

C --- Incident energies ---

      CALL HBOOK1(1,'INCIDENT KINETIC ENERGY$',1000,40.,2000.,0.)

C --- Start Positions ---

      CALL HBOOK2(2,'START POSITION XY',
     *  250,-1.E7,1.E7,250,-1.E7,1.E7,0.)
      CALL HBOOK2(3,'START POSITION YZ', 250,-1.E7,1.E7,250,0.,1.E7,0.)
      CALL HBOOK2(4,'START POSITION ZX', 250,0.,1.E7,250,-1.E7,1.E7,0.)

C --- Incident direction ---

      CALL HBOOK2(5,'INCIDENT DIRECTION$',
     *  181,-1.,361.,202,-1.01,1.01,0.)

C --- Particle number ---

      CALL HBOOK1(6,'RESULTING PARTICLES$',50,0.5,50.5,0.)
      END
```

A.2.9 sontel.inc

```
      COMMON/PROTONSPEC/ PROEKIN(50),PROSPECDIFF(50)
     *   ,PROSPECINT(50), NPRO
        COMMON/SELECTSPEC/ SELEKIN(50),SELSPECDIFF(50)
     *   ,SELSPECINT(50), NSEL
      COMMON/SPECHOICE/ISPECTRUM
      COMMON/POSINIT/ POSSTART(3)
      COMMON/DIRINIT/ PHISTART, THETASTART
      COMMON/COPYNR/ NCOPY
      COMMON/DIRPC/ EKININC
      COMMON/PCVETODIR/ ETOTCOUNTER(124), ETOTCOUNTERS,IPCG1,
     * IPCG2,IDPCG, IVETO,IDIR(5), EIONI
      COMMON/SCINTILLATOR/ ETOTSCINT(4), ETOTSCINTS, ICH1, ICH2,
     *  ICH3, ICH4, ISCIN, INIT, DISC(4)
      COMMON/SONDEBUG/ ISONTELDEBUG
      COMMON/SWITCHPC/ ISWITCHPC(2)
      COMMON/SOURCE/SDIRR(3),SDIRTH(3),SDIRPH(3),SOPHI,SOTHETA,
     *      SOCENTER(3),SODISTANCE,SORADIUS,ISOURCEON
      COMMON/DETECTORBOX/DETCENTER(3),DETSIZE(3),ICHECKCROSSING
      COMMON/AIRSTEP/SAIRDMAXMS,SAIRDEEMAX,SAIREPSIL,SAIRSTMIN
      COMMON/POLYSTEP/SPOLYDMAXMS,SPOLYDEEMAX,SPOLYEPSIL,
     *      SPOLYSTMIN
      COMMON/VACSTEP/SVACDMAXMS,SVACDEEMAX,SVACEPSIL,SVACSTMIN
      COMMON/GASSTEP/SGASDMAXMS,SGASDEEMAX,SGASEPSIL,SGASSTMIN
      COMMON/METSTEP/SMETDMAXMS,SMETDEEMAX,SMETEPSIL,SMETSTMIN
      COMMON/MAIN/NRTHETA,NRPHI,ZENITHINT,COSEXP,TIME,IFILENR
      COMMON/DETVOL/IDETL
```

A.3 Application SONTELSEP

A.3.1 main.F

```
C=====================================================================C
C MAIN.F - Part of the SONTELSEP GEANT3 application                   C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/26/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      PROGRAM SONTELSEPB
C---------------------------------------------------------------------C
C     Detector simulation with rotating source and solid angle        C
C     dependence of intensity                                         C
C=====================================================================C

#include "geant321/gcflag.inc"
#include "pvolum.inc"
#include "sontel.inc"

      COMMON/PAWC/H(1000000)
      COMMON/GCBANK/Q(5000000)

      INTEGER NRTHETA, NRPHI
      REAL ZENITHINT, COSEXP
      REAL DELTACOS, DELTAPHI, OMEGA, TIME, AREA
      REAL COSTHETA, THETA, PHI
      REAL PI
      REAL RINTENSITY(9,9)
      DATA RINTENSITY/81*0/
      PARAMETER(PI=3.1415926535)
      CHARACTER RUNNAME*6, HISTOID*4
      CHARACTER SPINTFILE*17
      INTEGER IPARTBIN(20)
      DATA IPARTBIN/1,2,3,0,4,5,0,6,7,0,0,0,8,9,9,0,0,0,0,0/
      REAL SPINT(7,6,25)
      DATA SPINT/1050*0./

      CALL TIMEST(1E5)
      CALL GZEBRA(5000000)
      CALL HLIMIT(-1000000)
      CALL UGINIT

C === Source definition ===

      DELTACOS=1./(NRTHETA-1)
      DELTAPHI=PI/2./(NRPHI-1)
      OMEGA=DELTACOS*DELTAPHI
      AREA=SORADIUS**2.*PI
      NRPHIMULT=4*(NRPHI-1)+1
      NRPART=5
      NPRODEVENTS = 0
      K=IPARTICLE

C --- Intensity ---

        DO M=0,NRTHETA-1,1
          DO N=0,NRPHIMULT-1,1
            WRITE(HISTOID,100)1000*IPARTBIN(K)+100*M+N
100         FORMAT(I4)
            SPINTFILE='spec/'//HISTOID//'.int'
```

```
            OPEN(10,FILE=SPINTFILE)
            READ(10,*)SPINT(K,M,N)
            CLOSE(10)
          ENDDO
        ENDDO

C --- Starts the runs for various positions ---

      DO M=0,NRTHETA-1,1
        DO N=0,NRPHIMULT-1,1
          CALL RESETHISTO
          WRITE(HISTOID,200)1000*IPARTBIN(K)+100*M+N
200       FORMAT(I4)
          CALL EXTERNAL_INTSPEC(HISTOID)
         COSTHETA=1.-M*DELTACOS
         THETA=ACOS(COSTHETA)*180./PI
         PHI=-90.+N*DELTAPHI*180./PI
          IF (M.NE.NRTHETA-1) THEN
            SPINT(K,M,N)=SPINT(K,M,N)/COSTHETA
          ELSE
            SPINT(K,M,N)=SPINT(K,M,N)/0.05
          ENDIF
          CALL USERRUN(THETA,PHI,INT(SPINT(K,M,N)*AREA*TIME))
          WRITE(RUNNAME,300)IFILENR
300       FORMAT(I5,'_')
          CALL STOREHISTO(RUNNAME//HISTOID)
        ENDDO
      ENDDO
      NPRODEVENTS = 0
      CALL RESETHISTO

      CALL UGLAST

      END



C=====================================================================C
      SUBROUTINE USERRUN(THETA,PHI,NOEVENTS)
C---------------------------------------------------------------------C
C     Positioning of the source and start of the simulation with a    C
C     certain number of events to be produced                         C
C=====================================================================C

      COMMON/PRODEVENTS/NPRODEVENTS
      INTEGER NOEVENTS

      NPRODEVENTS=0
      IF (ISOURCEON.NE.0) THEN
        CALL NEW_SOURCE_ORIENTATION(PHI,THETA)
      ENDIF
400   CALL GTRIGI
      CALL GTRIG
      CALL GTRIGC
      IF (NPRODEVENTS.LT.NOEVENTS) GOTO 400

      END
```

```
C==========================================================================C
      SUBROUTINE STOREHISTO(FILENAME)
C--------------------------------------------------------------------------C
C     Stores histograms                                                    C
C==========================================================================C

      CHARACTER FILENAME*9, FILEEXT*6, DIR*6

      DIR='hbook/'
      FILEEXT='.hbook'
      CALL HRPUT(0,DIR//FILENAME//FILEEXT,'T')

      END


C==========================================================================C
      SUBROUTINE RESETHISTO
C--------------------------------------------------------------------------C
C     Resets histograms                                                    C
C==========================================================================C

      DO I=1,16
        CALL HRESET(I,' ')
      ENDDO

      END
```

A.3.2 gukine.F

```
C=====================================================================C
C GUKINE.F - Part of the SONTELSEP GEANT3 application                 C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/22/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE GUKINE
C---------------------------------------------------------------------C
C     Generates Kinematics for primary track                          C
C=====================================================================C

#include "geant321/gcbank.inc"
#include "geant321/gcflag.inc"
#include "geant321/gckine.inc"
#include "geant321/gconsp.inc"
#include "geant321/gcscan.inc"
#include "celoss.inc"
#include "pvolum.inc"
#include "sontel.inc"

      CHARACTER*20 CHNPAR
      DIMENSION VERTEX(3),PLAB(3),RNDM(1),UB(10)
      DIMENSION NID1(200),NID2(10000),NID3(10000)
      DIMENSION NID4(10000),NID5(36000),SDIR(3)
      SAVE NID1, NID2,NID3, NID4,NID5
      DATA NID1/200*0/NID2/10000*0/NID3/10000*0/NID4/10000*0/
      DATA NID5/36000*0/
      SAVE VERTEX,PLAB
      DATA VERTEX/0.,1.,70./PLAB/3*0./
      COMMON/PRODEVENTS/NPRODEVENTS

      IF(SCANFL) THEN
         CALL GSCANK
      ELSE
         DEDL=0.
         EKINMAX=PKINE(2)
         IF (ISPECTRUM.NE.0) THEN
          EKIN=COMPUTE_ENERGY(EXTEKIN,EXTSPECDIFF,EXTSPECINT, NEXT)
         ELSE
          IF(IKINE.EQ.0)THEN
           EKIN=PKINE(1)
          ELSE
           CALL GRNDM(RNDM,1)
           EKIN=PKINE(1)+(PKINE(2)-PKINE(1))*RNDM(1)
          ENDIF
         ENDIF
         CALL GFPART(IPARTICLE,CHNPAR,ITRTYP,AMASS,CHARGE,TLIFE,UB,NWB)
         ETOT=EKIN+AMASS
         PLAB(3)=SQRT((ETOT+AMASS)*(ETOT-AMASS))
         IF (ISOURCEON.EQ.0) THEN
          VERTEX(1)=POSSTART(1)
          VERTEX(2)=POSSTART(2)
          VERTEX(3)=POSSTART(3)
          fac=3.1415926/180.
          phi=PHISTART*fac
          theta=THETASTART*fac
          PLAB(1)=PLAB(3)*SIN(theta)*COS(phi)
          PLAB(2)=PLAB(3)*SIN(theta)*SIN(phi)
          PLAB(3)=PLAB(3)*COS(theta)
```

```
         ELSE
   36    CALL CIRCULAR_SOURCE(VERTEX,SDIR)
          PLAB(1)=PLAB(3)*SDIR(1)
          PLAB(2)=PLAB(3)*SDIR(2)
          PLAB(3)=PLAB(3)*SDIR(3)
          fac=3.1415926/180.
          phi=sophi*fac
          theta=sotheta*fac
          ICONDITION=1
          IF (ICHECKCROSSING.NE.0)
     *        ICONDITION=ICROSSING_WITH_DETECTOR(VERTEX,SDIR)
          IF (ICONDITION.EQ.0) THEN
            IEVENT=IEVENT+1
            NPRODEVENTS=NPRODEVENTS+1
            GOTO 36
          ENDIF
         ENDIF

         NPRODEVENTS=NPRODEVENTS+1
         CALL GSVERT(VERTEX,0,0,0,0,NVERT)
         CALL GSKINE(PLAB,IPARTICLE,NVERT,0,0,NT)

C --- register primary particle ---

         EKININC=EKIN*1000.
         CALL HFF1(1,NID1,EKIN*1000.,1.)
         CALL HFF2(2,NID2,VERTEX(1),VERTEX(2),1.)
         CALL HFF2(3,NID3,VERTEX(2),VERTEX(3),1.)
         CALL HFF2(4,NID4,VERTEX(3),VERTEX(1),1.)
         CALL HFF2(5,NID5,phi/fac,COS(theta),1.)

C --- initialisation GUSTEP ---

         IVETO=1
         IDIR0SN=1
         IDIR25S=1
         IDIR25N=1
         IDIR40S=1
         IDIR40N=1
         IDIR(1)=1
         IDIR(2)=1
         IDIR(3)=1
         IDIR(4)=1
         IDIR(5)=1

         DO I=1,124
          ETOTCOUNTER(I) = 0
         ENDDO
          ETOTCOUNTERS   = 0
         DO I=1,4
          ETOTSCINT(I)  = 0
         ENDDO
         ETOTSCINTS  =  0
         ICH1=1
         ICH2=1
         ICH3=1
         ICH4=1

C --- kinematics debug (controlled by ISWIT(1)) ---

         IF(IDEBUG.EQ.1) THEN
            IF(ISWIT(1).EQ.1) THEN
               CALL GPRINT('VERT',0)
               CALL GPRINT('KINE',0)
```

```
           ENDIF
         ENDIF
      ENDIF
      END


C=====================================================================C
      SUBROUTINE EXTERNAL_INTSPEC(FILENAME)
C---------------------------------------------------------------------C
C     Normalizes integral and differential external spectrum          C
C=====================================================================C

#include "sontel.inc"

      CHARACTER FILENAME*4,SPECFILE*13
      DIMENSION  DATAEKIN(1000), DATASPEC(1000)
      DATA DATAEKIN/1000*0./DATASPEC/1000*0./
      EMIN=40.
      EMAX=1990.

      SPECFILE='spec/'//FILENAME//'.txt'
      OPEN(10,FILE=SPECFILE)
      DO I=1,3
        READ(10,*)DATASPEC(I)
      ENDDO
      DO I=1,197
        DATAEKIN(I)=25+10.*I
        READ(10,*)DATASPEC(I)
      ENDDO
      CLOSE(10)

      NEXT=197
      CALL COMPUTE_INTSPEC(DATAEKIN,DATASPEC,EXTEKIN,EXTSPECDIFF,
     *                     EXTSPECINT,NEXT, EMIN, EMAX)
      END


C=====================================================================C
      FUNCTION COMPUTE_ENERGY(EK,SPECDIFF,SPECINT,NPOINT)
C---------------------------------------------------------------------C
C     Computes initial energy                                         C
C=====================================================================C

#include "sontel.inc"

      DIMENSION RNDM(1),EK(1000), SPECDIFF(1000), SPECINT(1000)
200   CALL GRNDM(RNDM,1)
      DO i=2, NPOINT
        IF ((RNDM(1).GE.SPECINT(i-1))
     *                   .AND.(RNDM(1).LE.SPECINT(i))) THEN
        x0=EK(i-1)
        x1=EK(i)
        A0=SPECDIFF(i-1)
        A1=SPECDIFF(i)
        EKIN=x0+(RNDM(1)-SPECINT(i-1))/(SPECINT(i)-SPECINT(i-1))*(x1-x0)
        EKIN=x1
        COMPUTE_ENERGY=EKIN/1000.
        IF (COMPUTE_ENERGY.GT.EKINMAX) GOTO 200
        RETURN
        ENDIF
      ENDDO
      END
```

```
C=====================================================================C
      SUBROUTINE COMPUTE_INTSPEC(DATAEK,DATASPEC,EK,SPECDIFF,
     *                                SPECINT,NPOINT, EMIN,EMAX)
C---------------------------------------------------------------------C
C     Computes integral spectrum                                      C
C=====================================================================C

#include "sontel.inc"
      DIMENSION  DATAEK(1000), DATASPEC(1000), EK(1000), SPECDIFF(1000)
      DIMENSION  SPECINT(1000)

      EK(1)=EMIN
      x=EMIN
      x0=DATAEK(1)
      x1=DATAEK(2)
      A0=DATASPEC(1)
      A1=DATASPEC(2)
      SPECDIFF(1)=A0+x*(A1-A0)/(x1-x0)
      EK(NPOINT)=EMAX
      x=EMAX
      x0=DATAEK(NPOINT-1)
      x1=DATAEK(NPOINT)
      A0=DATASPEC(NPOINT-1)
      A1=DATASPEC(NPOINT)
      SPECDIFF(NPOINT)=0.
      DO i=2, NPOINT-1
        SPECDIFF(i)=DATASPEC(i)
        EK(i)=DATAEK(i)
      ENDDO
      SPECINT(1)=0.
      DO i=2, NPOINT
        x0=EK(i-1)
        x1=EK(i)
        A0=SPECDIFF(i-1)
        A1=SPECDIFF(i)
        SPECINT(i)=SPECINT(i-1)+(x1-x0)*A1
        IF (A0-A1.LT.0.1) SPECINT(i)=SPECINT(i)+1.
        PRINT*,i,SPECINT(i-1),SPECINT(i)
      ENDDO
      DO i=1, NPOINT
       SPECDIFF(i)=SPECDIFF(i)/SPECINT(NPOINT)
       SPECINT(i)=SPECINT(i)/SPECINT(NPOINT)
      ENDDO
      END


C=====================================================================C
      SUBROUTINE CIRCULAR_SOURCE(POS,DIR)
C---------------------------------------------------------------------C
C     Computes position distribution of the circular source           C
C=====================================================================C

#include "sontel.inc"

      DIMENSION POS(3),DIR(3),RNDM(1)

      CALL GRNDM(RNDM,1)
      radius=SORADIUS*SQRT(RNDM(1))
      CALL GRNDM(RNDM,1)
      phi=3.1415926*2.*RNDM(1)
      COSPHI=COS(phi)
      SINPHI=SIN(phi)
```

```
      DO i=1,3
       POS(I)=SOCENTER(I)+SODISTANCE*SDIRR(I)
     *              +RADIUS*(COSPHI*SDIRTH(I)+SINPHI*SDIRPH(I))
       DIR(I)=-SDIRR(I)
      ENDDO
      END

C=====================================================================C
      SUBROUTINE NEW_SOURCE_ORIENTATION(PHI,THETA)
C---------------------------------------------------------------------C
C     Computes orientation of the circular source                     C
C=====================================================================C

#include "sontel.inc"

      SOPHI=PHI
      SOTHETA=THETA
      DEGRAD=3.1415926/180.
      COSPHI=COS(PHI*DEGRAD)
      SINPHI=SIN(PHI*DEGRAD)
      COSTH=COS(THETA*DEGRAD)
      SINTH=SIN(THETA*DEGRAD)
      SDIRR(1)=COSPHI*SINTH
      SDIRR(2)=SINPHI*SINTH
      SDIRR(3)=COSTH
      SDIRTH(1)=COSPHI*COSTH
      SDIRTH(2)=SINPHI*COSTH
      SDIRTH(3)=-SINTH
      SDIRPH(1)=-SINPHI
      SDIRPH(2)=COSPHI
      SDIRPH(3)=0
      END

C=====================================================================C
      FUNCTION ICROSSING_WITH_DETECTOR(POS,DIR)
C---------------------------------------------------------------------C
C     Checks if the particle will cross the detector                  C
C=====================================================================C

#include "sontel.inc"

      DIMENSION POS(3),DIR(3)

      ICROSSING_WITH_DETECTOR=0
      PREC=0.0000000000001
      TMIN=-1000./PREC
      TMAX=1000./PREC
      DO I=1,3
       P=POS(I)-DETCENTER(I)
      IF (ABS(DIR(I)).GE.PREC) THEN
       T1=(-DETSIZE(I)/2. - P)/DIR(I)
       T2=(DETSIZE(I)/2. - P)/DIR(I)
       TPMIN=MIN(T1,T2)
       TPMAX=MAX(T1,T2)
       TMIN=MAX(TMIN,TPMIN)
       TMAX=MIN(TMAX,TPMAX)
       IF (TMIN.GT.TMAX) RETURN
      ELSE
       IF (ABS(P).GT.(DETSIZE(I)/2.)) RETURN
      ENDIF
      ENDDO
      ICROSSING_WITH_DETECTOR=1
      RETURN
      END
```

A.3.3 uginit.F

```
C=====================================================================C
C UGINIT.F - Part of the SONTELSEP GEANT3 application                 C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                        C
C Last modification: 08/22/2002                                       C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UGINIT
C---------------------------------------------------------------------C
C     Routine to initialize GEANT/USER program and read data cards    C
C=====================================================================C

#include "geant321/gcunit.inc"
#include "geant321/gclist.inc"
#include "geant321/gckine.inc"
#include "pvolum.inc"
#include "celoss.inc"
#include "sontel.inc"

C --- Opens user files ---

      CALL UFILES

C --- Initializes GEANT ---

      CALL GINIT

      IPARTICLE=14
      CALL FFKEY('PART',IPARTICLE,1,'INTEGER')
      CALL FFKEY('SRCON',ISOURCEON,1,'INTEGER')
      CALL FFKEY('CENTER',SOCENTER,3,'REAL')
      CALL FFKEY('DISTANCE',SODISTANCE,1,'REAL')
      CALL FFKEY('SORAD',SORADIUS,1,'REAL')
      CALL FFKEY('SPHI',PHI,1,'REAL')
      CALL FFKEY('STHETA',THETA,1,'REAL')
      CALL FFKEY('PHI',PHISTART,1,'REAL')
      CALL FFKEY('THETA',THETASTART,1,'REAL')
      CALL FFKEY('POSI',POSSTART,3,'REAL')
      CALL FFKEY('IONI',EIONI,1,'REAL')
      CALL FFKEY('DISCRI',DISC,4,'REAL')
      CALL FFKEY('SONDBG',ISONTELDEBUG,1,'INTEGER')
      CALL FFKEY('SWITCHPC',ISWITCHPC,2,'INTEGER')
      CALL FFKEY('SPECTRUM',ISPECTRUM,1,'INTEGER')
      CALL FFKEY('DETCENT',DETCENTER,3,'REAL')
      CALL FFKEY('DETSIZ',DETSIZE,3,'REAL')
      CALL FFKEY('CROSSDET',ICHECKCROSSING,1,'INTEGER')
      CALL FFKEY('NRTHETA',NRTHETA,1,'INTEGER')
      CALL FFKEY('NRPHI',NRPHI,1,'INTEGER')
      CALL FFKEY('ZENITHINT',ZENITHINT,1,'REAL')
      CALL FFKEY('COSEXP',COSEXP,1,'REAL')
      CALL FFKEY('TIME',TIME,1,'REAL')
      CALL FFKEY('FILENR',IFILENR,4,'INTEGER')
      CALL FFKEY('VDMAXMS',SVACDMAXMS,1,'REAL')
      CALL FFKEY('VDEEMAX',SVACDEEMAX,1,'REAL')
      CALL FFKEY('VEPSIL',SVACEPSIL,1,'REAL')
      CALL FFKEY('VSTMIN',SVACSTMIN,1,'REAL')
      CALL FFKEY('ADMAXMS',SAIRDMAXMS,1,'REAL')
      CALL FFKEY('ADEEMAX',SAIRDEEMAX,1,'REAL')
      CALL FFKEY('AEPSIL',SAIREPSIL,1,'REAL')
      CALL FFKEY('ASTMIN',SAIRSTMIN,1,'REAL')
```

```
      CALL FFKEY('PDMAXMS',SPOLYDMAXMS,1,'REAL')
      CALL FFKEY('PDEEMAX',SPOLYDEEMAX,1,'REAL')
      CALL FFKEY('PEPSIL',SPOLYEPSIL,1,'REAL')
      CALL FFKEY('PSTMIN',SPOLYSTMIN,1,'REAL')
      CALL FFKEY('GDMAXMS',SGASDMAXMS,1,'REAL')
      CALL FFKEY('GDEEMAX',SGASDEEMAX,1,'REAL')
      CALL FFKEY('GEPSIL',SGASEPSIL,1,'REAL')
      CALL FFKEY('GSTMIN',SGASSTMIN,1,'REAL')
      CALL FFKEY('MDMAXMS',SMETDMAXMS,1,'REAL')
      CALL FFKEY('MDEEMAX',SMETDEEMAX,1,'REAL')
      CALL FFKEY('MEPSIL',SMETEPSIL,1,'REAL')
      CALL FFKEY('MSTMIN',SMETSTMIN,1,'REAL')
      CALL FFKEY('RNDM',NRNDM,2,'INTEGER')

      IKINE=1
      PKINE(1)=0.001
      PKINE(2)=0.01

      CALL GFFGO
      CALL GZINIT
      CALL GPART

      PRINT*,IPARTICLE

C --- Prints version number ---

      WRITE(LOUT,10000)

C --- Geometry and materials description ---

      CALL UGEOM
      CALL GLOOK('MATE',LPRIN,NPRIN,IM)
      CALL GLOOK('TMED',LPRIN,NPRIN,IT)
      CALL GLOOK('VOLU',LPRIN,NPRIN,IV)
      IF(IM.NE.0)CALL GPRINT('MATE',0)
      IF(IT.NE.0)CALL GPRINT('TMED',0)
      IF(IV.NE.0)CALL GPRINT('VOLU',0)

C --- Energy loss and cross-sections initializations ---

      CALL FLINIT
      CALL GMORIN
      CALL GPHYSI

C --- Position distribution initialization ---

        IF (ISOURCEON.NE.0) THEN
         CALL NEW_SOURCE_ORIENTATION(PHI,THETA)
        ENDIF

C --- Defines user histograms ---

      CALL UHINIT

C --- Copy number initialization ---

      NCOPY=0

10000 FORMAT(/,'  VERSION 1.00 ',/)

      END
```

A.3.4 uhinit.F

```
C=====================================================================C
C UHINIT.F - Part of the SONTELSEP GEANT3 application                  C
C---------------------------------------------------------------------C
C M. R. Moser, University of Bern, Switzerland                         C
C Last modification: 08/22/2002                                        C
C=====================================================================C

#include "geant321/pilot.h"

C=====================================================================C
      SUBROUTINE UHINIT
C---------------------------------------------------------------------C
C     Routine to book the user histograms                              C
C=====================================================================C

#include "geant321/gckine.inc"
#include "pvolum.inc"

C --- Histograms for shower development ---

      NBINS=BINLIMS(1)

C --- Incident energies ---

      CALL HBOOK1(1,'INCIDENT KINETIC ENERGY$',200,0.,2000.,0.)

C --- Start Positions ---

      CALL HBOOK2(2,'START POSITION XY',250,-500,500,250,-500,500,0.)
      CALL HBOOK2(3,'START POSITION YZ',250,-500,500,250,-500,500,0.)
      CALL HBOOK2(4,'START POSITION ZX',250,-500,500,250,-500,500,0.)

C --- Incident direction ---

      CALL HBOOK2(5,'INCIDENT DIRECTION$',
     *  181,-1.,361.,202,-1.01,1.01,0.)

C --- Proportional Counter Histograms ---

      CALL HBOOK1(6,'PCOUNTER VETO SIGNAL$',2,-0.5,1.5,0.)
      CALL HBOOK1(7,'ESTIMATED DIRECTION WITH VETO$',5,-2.5,2.5,0.)
      CALL HBOOK1(8,'ESTIMATED DIRECTION WITHOUT VETO$',
     *  5,-2.5,2.5,0.)
      CALL HBOOK2(12,'INCIDENT ENERGY OF DETECTED WV$',
     *  1000,40.,2000.,5,0.5,5.5,0.)
      CALL HBOOK2(13,'INCIDENT ENERGY OF DETECTED WOV$',
     *  1000,40.,2000.,5,0.5,5.5,0.)
      CALL HBOOK2(15,'INCIDENT ENERGY OF SCINTILLATING PARTICLE WV$',
     *  1000,40.,2000.,4,0.5,4.5,0.)
      CALL HBOOK2(16,'INCIDENT ENERGY OF SCINTILLATING PARTICLE WOV$',
     *  1000,40.,2000.,4,0.5,4.5,0.)
      CALL HBOOK1(14,'NUMBER OF DETECTED DIRECTIONS',5,0.5,5.5,0.)

C --- Scintillator Histogram ---

      CALL HBOOK1(9,'SCINTILLATOR DEPOSIT ENERGY$',2000,0.,2000.,0.)
      CALL HBOOK1(10,'SCINTILLATOR CHANNEL WITH VETO$', 4,.5,4.5,0.)
      CALL HBOOK1(11,'SCINTILLATOR CHANNEL WITHOUT VETO$',
     *  4,.5,4.5,0.)

      END
```

A.3.5 sontel.inc

```
      COMMON/EXTSPEC/ EXTEKIN(1000),EXTSPECDIFF(1000)
     *  ,EXTSPECINT(1000), NEXT
      COMMON/ELECTRONSPEC/ ELEEKIN(50),ELESPECDIFF(50)
     *  ,ELESPECINT(50), NELE
      COMMON/NEUTRONSPEC/ NEUEKIN(50),NEUSPECDIFF(50)
     *   ,NEUSPECINT(50), NNEU
      COMMON/NEUTRONSPEC2/ NEUEKIN2(50),NEUSPECDIFF2(50)
     *   ,NEUSPECINT2(50), NNEU2
      COMMON/MUONSPEC/ MUOEKIN(50),MUOSPECDIFF(50)
     *   ,MUOSPECINT(50), NMUO
      COMMON/PROTONSPEC/ PROEKIN(50),PROSPECDIFF(50)
     *   ,PROSPECINT(50), NPRO
      COMMON/GAMMASPEC/ GAMEKIN(50),GAMSPECDIFF(50)
     *   ,GAMSPECINT(50), NGAM
       COMMON/SELECTSPEC/ SELEKIN(1000),SELSPECDIFF(1000)
     *   ,SELSPECINT(1000), NSEL
      COMMON/SPECHOICE/ISPECTRUM
      COMMON/POSINIT/ POSSTART(3)
      COMMON/DIRINIT/ PHISTART, THETASTART
      COMMON/COPYNR/ NCOPY
      COMMON/DIRPC/ EKININC
      COMMON/PCVETODIR/ ETOTCOUNTER(124), ETOTCOUNTERS,IPCG1,
     * IPCG2,IDPCG, IVETO,IDIR(5), EIONI
      COMMON/SCINTILLATOR/ ETOTSCINT(4), ETOTSCINTS, ICH1, ICH2,
     *  ICH3, ICH4, ISCIN, INIT, DISC(4)
      COMMON/SONDEBUG/ ISONTELDEBUG
      COMMON/SWITCHPC/ ISWITCHPC(2)
      COMMON/SOURCE/SDIRR(3),SDIRTH(3),SDIRPH(3),SOPHI,SOTHETA,
     *      SOCENTER(3),SODISTANCE,SORADIUS,ISOURCEON
      COMMON/DETECTORBOX/DETCENTER(3),DETSIZE(3),ICHECKCROSSING
      COMMON/AIRSTEP/SAIRDMAXMS,SAIRDEEMAX,SAIREPSIL,SAIRSTMIN
      COMMON/POLYSTEP/SPOLYDMAXMS,SPOLYDEEMAX,SPOLYEPSIL,
     *      SPOLYSTMIN
      COMMON/VACSTEP/SVACDMAXMS,SVACDEEMAX,SVACEPSIL,SVACSTMIN
      COMMON/GASSTEP/SGASDMAXMS,SGASDEEMAX,SGASEPSIL,SGASSTMIN
      COMMON/METSTEP/SMETDMAXMS,SMETDEEMAX,SMETEPSIL,SMETSTMIN
      COMMON/MAIN/NRTHETA,NRPHI,ZENITHINT,COSEXP,TIME,IFILENR
      COMMON/EKINMAX/EKINMAX
```