

179700090



ENTE PER LE NUOVE TECNOLOGIE,  
L'ENERGIA E L'AMBIENTE

Dipartimento Innovazione

ISSN 1120-5571



IT9700090

# EGS4CYL UN CODICE PER SIMULAZIONI MONTECARLO DI UN APPARATO PET AD ALTA RISOLUZIONE SPAZIALE

S. FERRIANI, M. GALLI

Centro Ricerche "Ezio Clementi", Bologna

D. BOLLINI

Università degli Studi di Bologna  
Istituto di Fisica ed INFN - Sezione di Bologna

RT/INN/95/17

1 2 200

**Testo pervenuto nel novembre 1995**

**I contenuti tecnico-scientifici dei rapporti tecnici dell'ENEA  
rispecchiano l'opinione degli autori e non necessariamente quella dell'Ente.**

## **SUMMARY**

*This report describes a Montecarlo simulation method for the simulation of a Pet or Spect equipment. The method is based on the Egs4cyl code. This work has been done in the framework of the Hirespet collaboration, for the developing of an high spatial resolution tomograph, the method will be used for the project of the thomograph. The treated geometry consists of a set of coaxial cylinders, surrounded by a ring of detectors. The detectors have a box shape, a collimator in front of each of them can be included, by means of geometrical constraints to the incident particles. An isotropic source is in the middle of the system. For the particles transport the Egs4code is used, for storing and plotting results the Cern packages Higz and Hbook are used.*

## **RIASSUNTO**

Questo rapporto descrive metodi di simulazione Montecarlo di un apparato Pet o Spect. Il metodo è basato sul codice Egs4cyl, sviluppato nell'ambito della collaborazione Hirespet, per lo sviluppo di un tomografo ad alta risoluzione spaziale, e verrà utilizzato per la progettazione del tomografo. L'apparato tomografico è rappresentato in modo semplice da un insieme di cilindri concentrici, di materiali e spessori variabili, circondati da un anello di rivelatori. I rivelatori hanno forma di piastrelle di dimensione variabile; la presenza di un collimatore avanti ai rivelatori è rappresentata da limiti geometrici imposti alle traiettorie delle particelle che entrano nel rivelatore. Una sorgente di positroni isotropa è posta al centro del sistema. Per il trasporto di particelle il codice utilizza le routines di Egs4, un programma per la simulazione Montecarlo del trasporto di elettroni, positroni e gamma sviluppato a Stanford negli anni 80; per la memorizzazione e visualizzazione dei risultati sono utilizzate routines dei pacchetti Higz ed Hbook del Cern.

## Indice

1. INTRODUZIONE .....	1
2. IL CODICE EGS4CYL .....	1
3. TRASPORTO DI PARTICELLE.....	2
4. USO DEL CODICE.....	3
4.1 File di input .....	5
4.2 File di Output .....	6
4.3 Procedure per l'uso del Codice .....	7
5. STRUTTURA DEL CODICE .....	7
6. SIMULAZIONE DI UN SEMPLICE APPARATO .....	9
7. CONCLUSIONI .....	11
<b>APPENDICE A .....</b>	<b>12</b>
<i>cyl.dat</i> .....	12
<i>yaps.dat</i> .....	13
<i>histcyl.dat</i> .....	13
<b>APPENDICE B .....</b>	<b>16</b>
<b>APPENDICE C .....</b>	<b>19</b>
<b>BIBLIOGRAFIA .....</b>	<b>22</b>

## 1. INTRODUZIONE

Recentemente la disponibilità di nuovi scintillatori rapidi, ad alta densità, e facilmente lavorabili, come il YAP:Ce<sup>1</sup>, ha dato impulso a ricerche tese al miglioramento delle tecniche PET e SPECT, allo scopo di ottenere una più alta risoluzione spaziale in questo tipo di indagini mediche.

In quest'ambito è nata la collaborazione HIRES PET, che coinvolge, fra l'altro, l'ENEA, l'INFN, le Università di Bologna, Ferrara, Roma III e Roma "La Sapienza". L'idea è quella di abbinare uno scintillatore fatto di sottili bacchette di YAP ad un rivelatore ad alta risoluzione spaziale, per ottenere un rivelatore capace di risoluzioni attorno al millimetro in SPECT e di poco peggiori in PET, avvicinandosi a quelle che costituiscono il limite fisico di questi metodi<sup>1</sup>.

Questo aumento di risoluzione spaziale richiede la messa a punto di nuove tecniche di ricostruzione di immagini. Per la progettazione del tomografo occorre poi uno strumento in grado di simulare l'apparato e calcolare l'efficienza dei rivelatori in funzione della geometria del sistema. Questo strumento è il codice EGS4CYL che si descrive in questo rapporto.

EGS4CYL verrà anche utilizzato per generare set di eventi per il test e la messa a punto dei programmi di ricostruzione.

## 2. IL CODICE EGS4CYL

Il codice EGS4CYL è un Montecarlo basato su EGS4<sup>2</sup>, un package di uso generale per simulazione Montecarlo di trasporto di elettroni, positroni e fotoni, sviluppato a Stanford negli anni 70 ed 80 per la fisica delle alte energie; EGS4 è ora estensivamente usato anche in fisica medica, per simulare il comportamento di vari apparati diagnostici.

EGS4 consiste di:

- un pre-processore: PEGS4, che calcola sezioni d'urto e "branching ratios" in formato adatto ad EGS4, per i materiali e le misture di materiali richiesti dall'utente;
- una serie di routines: EGS4, che descrivono l'interazione degli elettroni e fotoni con i materiali.

EGS4 non contiene routines di input output, né routines che descrivono la geometria del sistema, queste routines, ed il "main program" devono essere scritte dall'utente. Il fatto che le routine geometriche siano scritte dall'utente è un grosso vantaggio per geometrie semplici, si riesce infatti ad avere un codice più veloce di quanto non si abbia con Montecarlo che sfruttano routines geometriche standardizzate, che in genere implementano una geometria combinatoria.

Le routines geometriche e di output sono la HOWFAR e la AUSGABE. AUSGABE, viene chiamata da EGS4 per l'output ogni volta che si verificano certi eventi definiti dall'utente; HOWFAR, routine geometrica, viene chiamata da EGS4 ogni volta che una particella si

sposta, ed ha il compito di calcolare se la traiettoria porterà la particella ad attraversare il confine fra due diverse zone del sistema.

Il codice EGS4CYL consiste quindi di :

- un "main program", che inizializza le variabili, legge l'input, definisce le caratteristiche della sorgente, chiama le routine di EGS4 per il trasporto di particelle, ed alla fine produce un output sintetico;
- un insieme di routines geometriche chiamate dalla HOWFAR;
- una routine AUSGABE che si serve delle routine del package HBOOK del CERN per immagazzinare dati ed istogrammi;
- routine grafiche, basate su HIGHZ, che durante l'esecuzione del programma visualizzano traiettorie delle particelle ed istogrammi

La geometria trattata dal codice é una schematizzazione semplice dell'insieme costituito dall'oggetto dell'analisi tomografica e dal rivelatore: una sorgente isotropa di positroni é posta al centro di una serie di cilindri concentrici, di materiali e spessori variabili, il tutto é circondato da un anello di rivelatori. I rivelatori hanno forma di piastrelle di dimensione variabile; la presenza di un collimatore avanti ai rivelatori é rappresentata da limiti geometrici imposti alle traiettorie delle particelle che entrano nel rivelatore.

L'input del codice consiste di una serie di file che specificano l'energia dei positroni della sorgente, la geometria del sistema, le caratteristiche dei file di output, il numero di eventi da simulare etc., é inoltre necessaria la libreria di dati sui materiali utilizzati, come fornita dal codice PEGS4, il "pre-processor" del codice EGS4.

Il codice produce una serie di file in formato "ZEBRA", ad accesso diretto, con istogrammi e dati sulla distribuzione dell'energia depositata nei vari cilindri e nel rivelatore, adatti per essere analizzati col programma PAW del CERN. Il programma produce anche "n-tuple" contenenti dati sul decadimento dei positroni e, per ogni storia, la distribuzione dell'energia nei singoli rivelatori.

Per la memorizzazione e visualizzazione dei risultati sono utilizzate routines dei pacchetti HIGZ ed HBOOK del CERN.

Il codice EGS4CYL, scritto in FORTRAN 77, é stato testato su macchine con sistema operativo VMS ed architettura VAX o DEC-ALPHA. Richiede le librerie grafiche del CERN. Il tempo di CPU per storia dipende in modo essenziale dal numero di cilindri e di rivelatori considerati. Per situazioni semplici abbiamo tipicamente 1 secondo per storia su una Vaxtation 3100 e 0.1 secondo per storia su una Dec Alpha3000/400.

### **3. TRASPORTO DI PARTICELLE**

Il trasporto di particelle é effettuato dalle routines di EGS4 che seguono elettroni e positroni di energia fra qualche migliaio di GeV e 10 keV e fotoni da qualche migliaio di GeV a 1 keV. Esiste una soglia di energia definita dall'utente (PCUT per i fotoni ed ECUT per elettroni):

quando le particelle arrivano a questa energia non vengono più seguite e la loro energia viene depositata nel punto in cui si trovano quando scendono sotto la soglia.

Per quanto riguarda il trasporto di particelle cariche EGS4 tiene conto delle seguenti reazioni:

- perdita di energia delle particelle cariche per bremsstrahlung, con produzione di fotoni;
- annichilazione di positroni, in volo ed a riposo, con produzione di fotoni;
- scattering coulombiano delle particelle cariche da parte dei nuclei, viene utilizzata la teoria di Molière dello scattering multiplo; alla fine del tratto percorso dall'elettrone o positrone (step) viene calcolata la variazione di direzione nel moto della particella e si tiene conto dell'aumento di percorso dovuto a scattering multiplo, ma non a variazioni nella posizione finale dell'elettrone. Questa approssimazione impone un limite superiore alla lunghezza di un singolo step. I limiti di validità della teoria di Molière impongono anche un limite minimo alla lunghezza dello step.
- scattering Møller, fra 2 elettroni, e scattering Bahabha , fra positrone ed elettrone;
- le perdite di energia sotto la soglia definita dall'utente (AP per i fotoni e AE per elettroni) sono rappresentate come un assorbimento continuo lungo la traiettoria della particella. Queste reazioni di bassa energia comprendono: soft bremsstrahlung, scattering con  $e'$  ed  $e$  a bassa energia, eccitazione degli atomi, scattering elastico multiplo con atomi.

Per quanto riguarda il trasporto dei fotoni vengono considerati:

- produzione di coppie elettrone positrone;
- scattering Compton;
- effetto fotoelettrico, nella versione standard non si tiene conto di fluorescenza o elettroni dell'effetto Auer;
- può essere incluso lo scattering Rayleigh.

Per i dettagli dei formalismi adottati per le varie interazioni si rimanda al manuale di EGS4<sup>2</sup>

#### 4. USO DEL CODICE

Il codice legge l'input da una serie di file che devono essere preparati dall'utente.

In caso si scelga l'uso interattivo il calcolo viene interrotto dopo un certo numero di storie ed il controllo viene passato all'utente che esamina i risultati parziali e decide se continuare il calcolo.

In particolare all'utente viene presentato un menu principale che ha le seguenti voci:

<b>G:</b>	va avanti nel calcolo
<b>S:</b>	mostra il setup grafico
<b>C:</b>	cambia il setup grafico
<b>F:</b>	fine calcolo

L'opzione **G** continua il calcolo, mentre **F** termina l'esecuzione del programma. La voce **S** mostra il setup grafico corrente, vale a dire se siano state abilitate, o meno, le modalità **TRACKING** ed **HISTOGRAM**. La prima, se abilitata, visualizza, in una finestra del terminale grafico dal quale è stato lanciato il programma (X-terminal), le viste in pianta e laterali del set-up sperimentale (cilindro e rivelatori) unitamente alla traiettoria delle particelle, man mano sono calcolate. Con il modo **HISTOGRAM** abilitato, si possono visualizzare fino a quattro finestre indipendenti, ciascuna delle quali riporta l'istogramma di grandezze di interesse. L'output grafico del codice, è illustrato in Fig. 1.

La voce **C** attiva un sottomenu col quale è possibile abilitare o disabilitare le modalità **TRACKING** ed **HISTOGRAM**. In particolare, l'abilitazione del **TRACKING** può avvenire in due modi: modo normale (**O**: on), nel quale prima di visualizzare le successive tracce, quelle già esistenti sullo schermo sono rimosse; nel modo continuo (**C**: on cumulative) le traiettorie sono aggiunte a quelle già tracciate sul terminale. Una volta abilitata la modalità **HISTOGRAM** è necessario scegliere quali istogrammi visualizzare. Tramite l'opzione **S**: setting histogram identifiers, si possono associare, a ciascuna delle quattro finestre, il codice dell'istogramma da visualizzare. Una lista mostra gli istogrammi a disposizione. Impostando il codice **0** la finestra corrispondente non sarà visualizzata, e, se preesistente sullo schermo, rimossa.

Un output sintetico viene prodotto su terminale e contemporaneamente su un file aschi; i risultati veri e propri sono conservati in formato **ZEBRA**, su file ad accesso diretto, sotto forma di istogrammi ed n-tuple; adatti ad essere direttamente utilizzati dal programma di analisi **PAW**, del **CERN**.

Nella preparazione dell'input occorre tener presente che l'insieme dei cilindri concentrici è descritto in un sistema cartesiano destrorso, con l'asse **z** lungo l'asse dei cilindri.

I rivelatori hanno forma di parallelepipedi, sono posti su piani paralleli all'asse **z** (asse dei cilindri) ed ortogonali al piano **x-y**. Ad ogni rivelatore è associato un sistema di riferimento destrorso, con centro nel centro del rivelatore, asse **z** parallelo all'asse **z** dei cilindri (altezza

dei rivelatori), asse x in senso radiale alle superfici dei cilindri (spessore del rivelatore), asse y in senso tangenziale (larghezza dei rivelatori).

Il sistema consiste di tante zone quanti sono i cilindri concentrici (NCYL è il numero di zone), la zona NCYL+1 rappresenta la zona esterna ai cilindri, fra i cilindri ed i rivelatori; le regioni da NCYL+2 a NCYL+1+NYP rappresentano i rivelatori (NYP è il numero di rivelatori), la zona NCYL+NYP+2 è una regione fittizia, che rappresenta le fughe dal sistema, la zona NCYL+NYP+3 rappresenta gli assorbimenti da parte dei collimatori, la zona NCYL+NYP+4 le particelle che fuggono dai rivelatori e vengono scartate in base al parametro di input : ESCONO ( vedi più oltre ). Le dimensioni sono tutte espresse in cm, le energie in MeV.

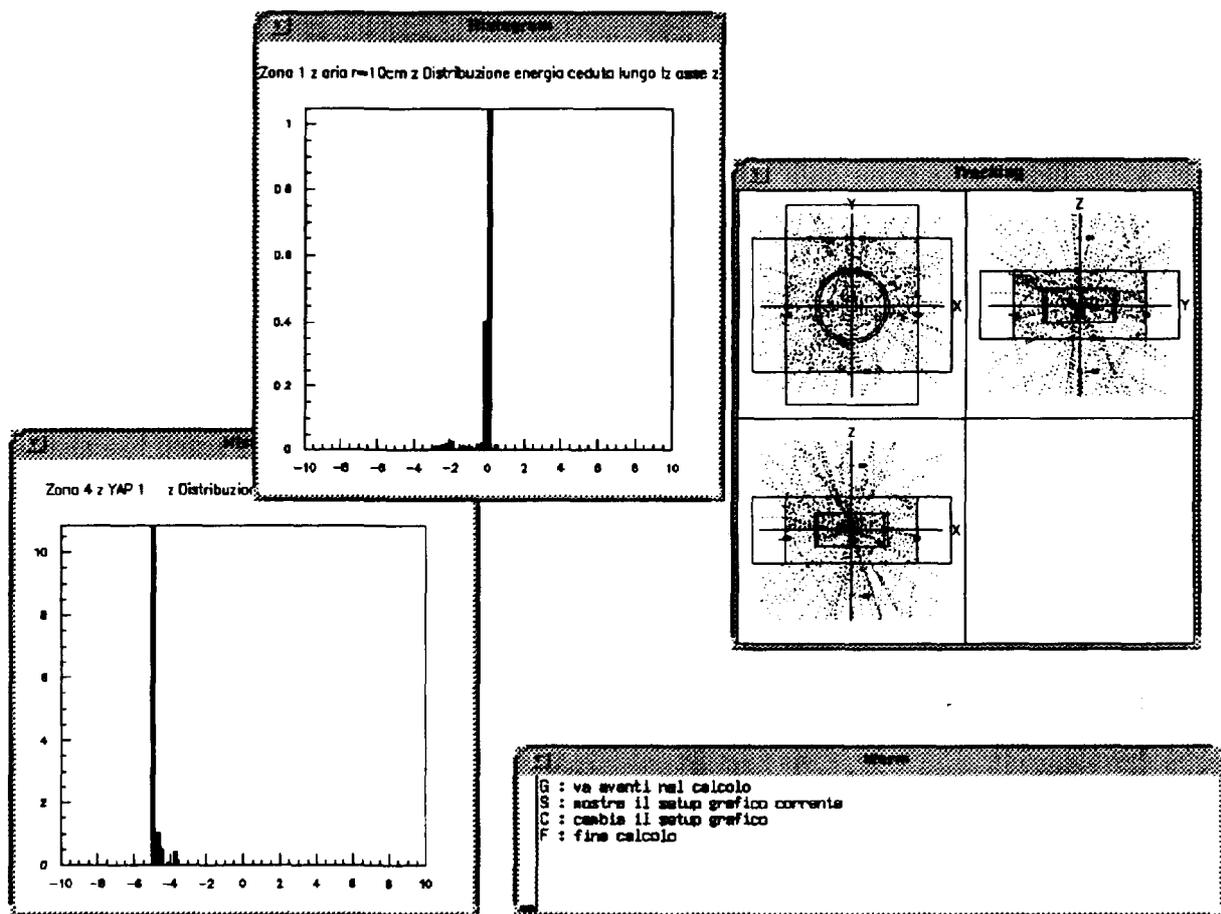


Figura 1 - Interfaccia grafica di controllo e visualizzazione intermedia dei dati

#### 4.1 File di input

I file utilizzati dal codice per l'input sono: MAT.DAT, CYL.DAT, YAPS.DAT e HISTCYL.DAT.

Il file MAT.DAT, di tipo ascii, è prodotto una volta per tutte dal pre-processore di EGS4: PEGS4; deve essere rigenerato solo nel caso si debbano aggiungere dati relativi a materiali non inclusi nel file stesso.

I file CYL.DAT, YAPS.DAT e HISTCYL.DAT, invece sono generati dall'utente e contengono, rispettivamente, dati sulla sorgente e la geometria, dati sulla geometria dei rivelatori e specifiche per gli istogrammi prodotti. La tabella 1 riassume le principali caratteristiche dei file di input, mentre una loro più dettagliata descrizione è riportata in Appendice A.

NOME FILE	NOME UNITÀ LOGICA	NUMERO	DESCRIZIONE
CYL.DAT	ltydat	7	specifiche per geometria e materiali
YAPS.DAT	n13	13	specifiche dei rivelatori
HISTCYL.DAT	ltyfil	9	specifiche per gli istogrammi prodotti
MAT.DAT	ltyin	12	file prodotto da Pegs4
-	ltyin	5	input da terminale

Tabella 1 - File di Input

#### 4.2 File di Output

Il codice utilizza diversi file per memorizzare i risultati dei calcoli effettuati. In particolare i file di output sono riportati nella tabella 2.

NOME FILE	NOME UNITÀ LOGICA	NUMERO	DESCRIZIONE
OUT.DAT	ltyhis	10	output sommario
HATCH.OUT		8	descrizione materiali
HBOOK.OUT	hislun	20	file formato Zebra con istogrammi
NTUPLE.OUT	hislun1	21	file formato Zebra con n-tuple
NTUPLE.Y.OUT	hisluny	22	file formato Zebra con n-tuple per rivelatori
PARTICLES.OUT	n50paths	50	traiettorie particelle
-	-	11	file postscript per plots
-	ltyout	6	output su terminale

Tabella 2 - File di Output

Il principale file di output è OUT.DAT, un file ascii, su 132 colonne, con una descrizione della geometria del sistema, dei rivelatori, dei materiali nelle singole zone, con gli istogrammi

richiesti in input ed un sommario della deposizione di energia nelle varie zone e nei diversi rivelatori.

Il file HATCH.DAT é prodotto dalla subroutine HATCH di EGS4 e contiene un sommario dei dati che la subroutine HATCH legge dal file MAT.DAT.

Il file HBOOK.OUT é un file ad accesso diretto, in formato ZEBRA, contenente gli istogrammi con la distribuzione di energia depositata nei cilindri e nei rivelatori.

Il file NTUPLE.OUT é un file ad accesso diretto, in formato ZEBRA, e contiene le n-tuple con le coordinate dei punti di decadimento dei positroni.

Il file NTUPLE.Y.OUT contiene le n-tuple relative agli eventi che si verificano nei rivelatori.

Il file PARTICLES.OUT é un file ascii, su 132 colonne, e contiene una descrizione delle traiettorie delle particelle, é un file molto grosso, usato essenzialmente per test, ed é prodotto solo dietro richiesta dell'utente ( parametro NOUTFLAG5  $\leq$  2).

### **4.3 Procedure per l'uso del Codice**

La versione attuale del codice é stata provata su macchine Digital, ( Alpha e Vax ) con sistema operativo VMS. Per l'uso del codice sono quindi state scritte procedure DCL che facilitano le principali operazioni per l'uso del codice: compilazione, linking ed esecuzione. Per quest'ultima operazione, la procedura si incarica di allocare i file di input ed output, e l'utente può intervenire su di essa per modificare i nomi dei file utilizzati. Esempi di tali procedure sono riportati in Appendice B

## **5. STRUTTURA DEL CODICE**

Il codice consta di una serie di subroutine da EGS4 per il trasporto delle particelle, e di routines per inizializzare le variabili, leggere l'input, scrivere i risultati, descrivere la geometria del sistema.

L'organizzazione delle varie parti segue quella che é la logica di EGS4. Un "main" chiama le routines che leggono l'input, poi, per ogni storia, la routine SOURCE\_CYL, che genera una particella iniziale, e la routine SHOWER (di EGS4) per il trasporto. In base a flag fissati dall'utente, le routines di EGS4 chiamano AUSGABE per l'output dei risultati e, prima di spostare una particella, la routine HOWFAR, che calcola se la traiettoria attraversa il confine fra due diverse zone del sistema. Alla fine il main provvede a calcolare la frazione di energia depositata in ogni zona ed altre quantità di interesse.

Questa logica é implementata in una serie di routines che vengono elencate e brevemente illustrate nella tabella 3.

FILE E ROUTINES DEL PROGRAMMA EGS4CYL			
FILE	ROUTINES	FUNZIONE	ROUTINE CHIAMATE
egs4_cyl_2.for	main program	lettura dell'input	- INIT_CYL - READ_YAPS - HISTIN_CYL - HATCH*
		tracking storie	- SOURCE_CYL - SHOWER *
		controllo risultati intermedi	- GUARDA
		memorizzazione finale dei dati	- routines HBOOK - STATEND_CYL * routines EGS4
common_cyltot_2.for		commons inclusi dalle routines	
comm_yaps.for		commons per la geometria dei rivelatori	
init_cyl.for	init_cyl	inizializza numeri unità logiche, legge CYL.DAT	
yaps.for	read_yaps	legge YAPS.DAT calcola dati sulla geometria dei rivelatori	- INYAPSYS - INCYLSYS - INCYLSYS0
histin_cyl_2.for	histin_cyl	apre i file di output, legge il file HISTCYL.DAT	
source_cyl.for	source_cyl	genera positroni nell'origine con distribuzione isotropa	
guarda_2.for	guarda	routines per il controllo interattivo	
howfar_cyl_2a.for	howfar	routine geometrica	- HOWCYLS (cilindri) - HOWYAPS (rivelatori)
	howcyls	geometria zona cilindri	- PARLA
	howyaps	geometria zona rivelatori	- GUARDAATT - ATTRAVX
	parla	messaggi diagnostici	
	guardaatt	intersezione segmento parallelepipedo	
	attravx	intersezione con la zona cilindri	
	intersec	intersezione retta-cilindro	
ausgabe_cyl_2a.for	ausgabe	prepara istogrammi in formato ZEBRA	- AUSGAB0 - AUSGABHFN
staden_cyl.for	staden_cyl	statistiche finali del calcolo	
petgraf.for		routines per la grafica ed il controllo interattivo	
petgraf.inc		incluso delle routines grafiche	- routine HIGZ
petgraf_com.inc		incluso per i commons delle routines grafiche	
petgraf_data.inc		incluso di inizializzazione commons routines grafiche	
petgraf_higz.inc		incluso routines grafiche di HIGZ	

Tabella 3 - Files e routines del programma EGS4CYL

## 6. SIMULAZIONE DI UN SEMPLICE APPARATO

Come esempio di uso del codice viene qui mostrata la simulazione di un semplice apparato, consistente di :

- un cilindro di aria, di raggio 2 cm e di semialtezza 2.1 cm (zona 1);
- il cilindro di aria é circondato da uno spessore di alluminio di 0.5 cm (zona 2);
- attorno al cilindro c'è aria (zona 3) attorno al cilindro sono disposti, ad anello poligonale, 6 rivelatori sotto forma di piastrelle di YAP, di 4.2\*4.2 cm spesse 2 cm (zone da 4 a 9);
- al centro del sistema é posta una sorgente di positroni da 1 Mev;
- in questo caso non é inserito un collimatore davanti ai rivelatori.

La zona 10 é una zona fittizia che rappresenta le fughe dal sistema, la zona 11 rappresenterebbe gli assorbimenti da parte del collimatore, la zona 12 rappresenta le particelle uscite dai rivelatori e non ulteriormente seguite nel calcolo.

Il comportamento del sistema é abbastanza intuitivo: i positroni emessi dalla sorgente termalizzano nell'alluminio, ove annichilano. Sui rivelatori arrivano i fotoni di annichilazione, che sono in parte assorbiti.

Vengono effettuate 40000 storie, il calcolo é stato effettuato su una Dec Alpha 3000/400, con sistema operativo VMS. Il tempo richiesto é stato di circa 1 ora (11 storie al secondo).

Il calcolo ci dice che circa 1/3 dei positroni fuggono assialmente dal sistema (come ci si poteva aspettare da semplici considerazioni geometriche), ed il resto termalizza ed annichila nell'alluminio.

Dell'energia iniettata nel sistema (compresa la massa dei positroni) circa il 46% é perso per fughe dal sistema, il 36% é depositato nell'alluminio e circa l'1% é assorbito da ogni rivelatore.

A fine calcolo, sul file CYL.OUT ed al terminale, é presentato un breve sommario dei risultati, così come illustrato in Appendice C assieme ai file di input utilizzati per il calcolo.

La figura 2 rappresenta la distribuzione x-y dell'energia depositata nell'apparato. É chiaramente visibile la zona di alluminio ove si annichilano i positroni e la forma dei 6 rivelatori.

La figura 3 riporta la distribuzione r-z dell'energia depositata nell'apparato, anche qui si vede come la maggior parte dell'energia sia depositata nello strato di alluminio.

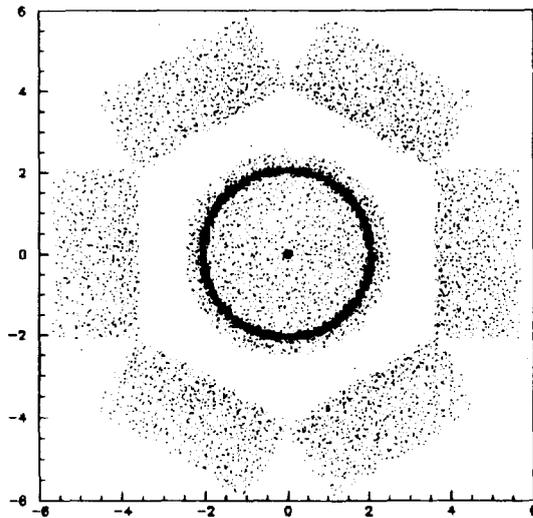


Fig. 2 - Distribuzione x-y dell'energia depositata

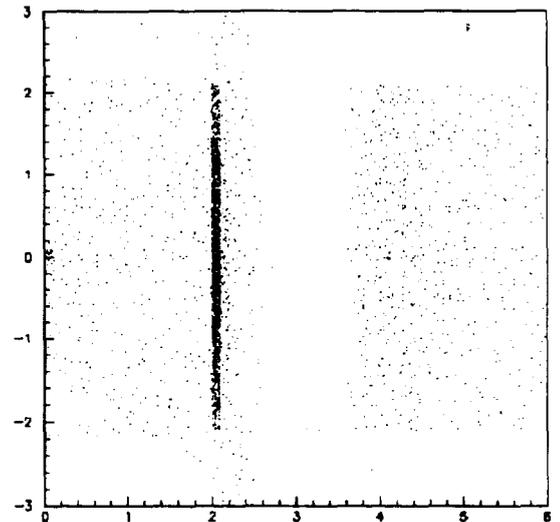


Fig. 3 - Distribuzione r-z dell'energia depositata

La figura 4 mostra i punti in cui decadono i positroni, praticamente tutti i positroni decadono nei primi 2 millimetri di alluminio.

La figura 5 illustra la distribuzione lungo l'asse z dell'energia persa nell'alluminio, la configurazione geometrica dell'apparato fa sì che le zone centrali del cilindro di alluminio ricevano più energia, l'andamento con z va come un coseno.

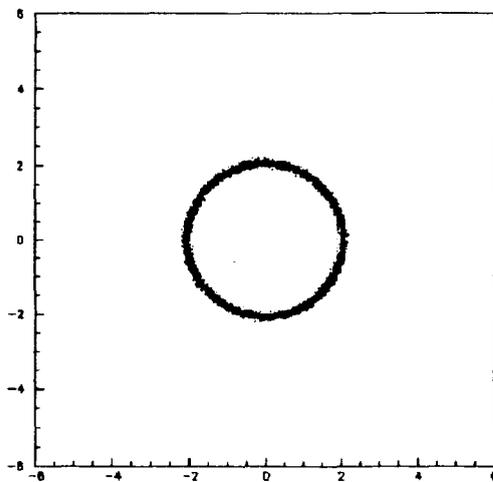


Fig. 4 - Distribuzione x-y dei punti di decadimento dei positroni

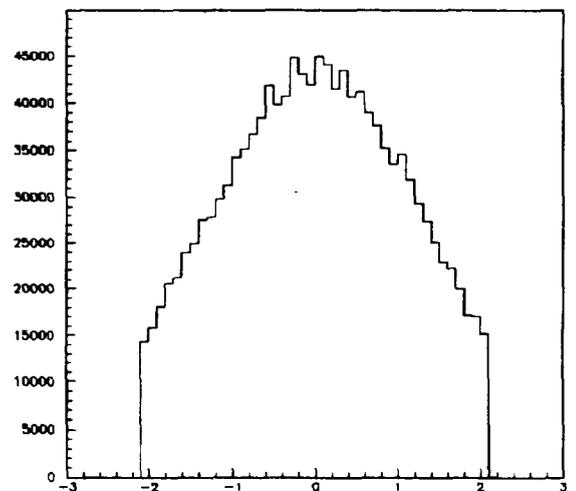


Fig. 5 - Energia persa nell'alluminio lungo l'asse z

La figura 6 mostra la distribuzione di energia in profondità in un rivelatore. Si vede uno spessore di 2 cm non sia sufficiente ad assorbire tutta l'energia incidente.

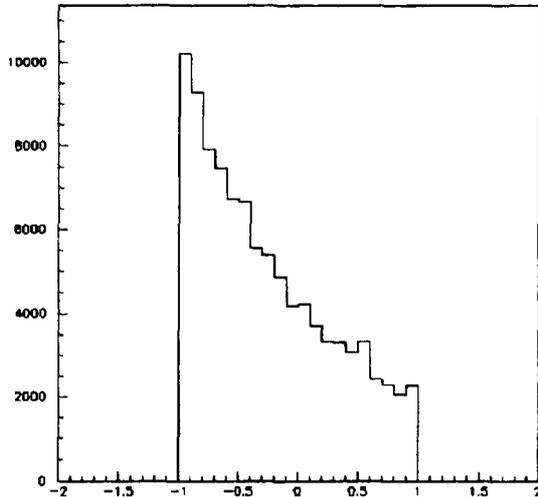


Fig. 6 - Distribuzione di energia in profondità in un rivelatore

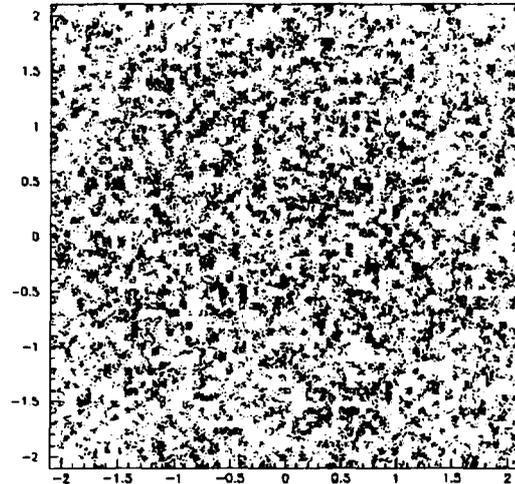


Fig. 7 - Distribuzione di energia sulla faccia dei rivelatori

La figura 7 riproduce la distribuzione di energia sulla faccia dei rivelatori.

## 7. CONCLUSIONI

Il codice EGS4CYL costituisce uno strumento per progettazione e simulazione di apparati tomografici versatile e di impiego relativamente semplice. Queste caratteristiche sono dovute anche alla parte interattiva e grafica del programma che consente un costante controllo dell'evoluzione della simulazione e dell'andamento dei parametri di interesse.

EGS4CYL, pur non essendo stato ottimizzato, ha delle prestazioni più che accettabili per un codice di questo tipo. In particolare le performance dipendono in modo critico dalla complessità della geometria descritta.

Infine, va sottolineato che la scelta di utilizzare le routine grafiche HIGZ, che offrono una interfaccia a X Window (X11), consente di eseguire il codice da macchine e terminali remoti, senza rinunciare alla parte grafica.

## Appendice A

Il codice EGS4CYL utilizza 3 file di input per definire sia i valori iniziali di parametri e variabili sia le caratteristiche dell'apparato da simulare. Di seguito sono descritti il contenuto e la struttura di tali file.

### cyl.dat

Il file CYL.DAT contiene dati richiesti da EGS4, come ECUT, PCUT, ICHAR, e dati che specificano la sorgente e la geometria. La sua struttura é descritta dalla seguente tabella.

CYL.DAT		
SCHEDA	FORMATO	CONTENUTO
1	*	NCASES      numero storie ICHAR       carica iniziale particella (-1 : elettroni; 1: positroni) INIREG      regione in cui si trova la sorgente EKIN        energia cinetica delle particella iniziale
2	*	ITCX        ogni Itcx storie vengono stampate varie informazioni, e passa il controllo all'utente ISTSEED    numero storia iniziale ( Istseed ≠ 0 il Montecarlo inizia utilizzando il seed di input) ISTARTSEED seed di input utilizzato se Istseed ≠ 0
3	*	NMED        numero materiali
4	24A1	MEDIA
5	*	MAT        numero del materiale PCUT        energia minima per fotoni ECUT        energia minima per elettroni se l'energia scende sotto Ecut o Pcut la particella si considera assorbita
6	*	NCYL        numero cilindri ZMAX        metà altezza dei cilindri
7	*	RTHICK     distanza, in senso radiale, dal cilindro piú interno INDMAT     numero del materiale nella zona fra questo cilindro e il precedente
8	*	NOUTFLAG1 Noutflag1 ≥ 0: uso interattivo del programma NOUTFLAG2 Noutflag2 > 0: vengono stampate informazioni sulle traiettorie particelle sul file n50path NOUTFLAG3 Noutflag3 > 0: vengono prodotti sul file hislun1 i dati sul decadimento dei positroni NOUTFLAG4 Noutflag4 > 0: vengono prodotti sul file hislun1 i dati sull'energia persa nei rivelatori NOUTFLAG5 Noutflag5 > 0: stampa sul terminale informazioni sulle traiettorie delle particelle Noutflag5 = 2: le informazioni sono prodotte solo sul file n50path Noutflag5 >2: le informazioni sono prodotte sia sul file sia su terminale NOUTFLAG6 non utilizzato

Tabella A1 - Schede del file CYL.DAT

## yaps.dat

Il file YAPS.DAT contiene la descrizione geometrica dei rivelatori. L'utente può dare le dimensioni e le posizioni dei singoli rivelatori oppure può far porre rivelatori tutti eguali ad anello attorno al cilindro e dare solo il numero di rivelatori, la distanza dall'asse del cilindro e la posizione dell'anello lungo l'asse del cilindro.

Occorre far attenzione quando si descrivono i singoli rivelatori perché il programma non controlla l'autoconsistenza dell'input, ed in particolare non controlla che i rivelatori non si intersechino o non intersechino i cilindri. Il contenuto del file è descritto nella tabella A2.

YAPS.DAT			
SCHEDA	FORMATO	CONTENUTO	
1	*	NYP	numero dei rivelatori - NYP > 0 i rivelatori sono descritti uno per uno e si devono fornire le schede numero 3 (vedi sotto) - NYP = 0 i rivelatori non sono
		RYP	distanza tra l'asse del cilindro e il centro della faccia interna dei rivelatori - RYP = 0 il programma pone NYP rivelatori adiacenti che formano un anello poligonale chiuso
		ZYP	posizione dei rivelatori in altezza, lungo l'asse del cilindro
		DXYP	spessore dei rivelatori (in direzione radiale)
		DYYP	larghezza dei rivelatori (in direzione tangente ai cilindri)
		DZYP	altezza dei rivelatori (nella direzione dell'asse del cilindro)
		MEZZO	numero del materiale della zona fra i cilindri ed i rivelatori
		MEZZOYAPS	numero del materiale dei rivelatori
2	*	ANGOLCOL	angolo massimo fra perpendicolare al rivelatore e particella entrante, le particelle con angolo maggiore sono assorbite dal collimatore
		ESCONO	ESCONO > 0 le particelle uscenti dai rivelatori sono assorbite nella zona fittizia numero NCYL+NYP+4 ESCONO < 0 le particelle possono uscire dai rivelatori ESCONO = 0 le particelle possono uscire dai rivelatori, ma non quelle della faccia verso i cilindri assorbite nella zona NCYL+NYP+4
3	*	RYPI	distanza tra l'asse del cilindro e il centro della faccia interna del rivelatore
		ZYPI	posizione del rivelatore in altezza, lungo l'asse del cilindro
		DXYPI	spessore del rivelatore (in direzione radiale)
		DYYPI	larghezza del rivelatore (in direzione tangente ai cilindri)
		DZYPI	altezza del rivelatore (nella direzione dell'asse del cilindro)
		MEZZOYPI	numero del materiale della zona fra i cilindri ed il rivelatore
		ANGYPI	angolo cui è posto il rivelatore, misurato dall'asse x verso asse x del rivelatore, in senso antiorario

Tabella A2 - Schede del file YAPS.DAT

## histcyl.dat

Il file HISTCYL.DAT contiene dati sugli istogrammi che si vuole il programma produca su file HBOOK.OUT, in formato ZEBRA, per l'analisi con PAW. Vengono prodotti istogrammi

monodimensionali e bidimensionali su file HBOOK.OUT; n-tuple sul file NTUPLE.OUT, per quanto riguarda le posizioni di decadimento dei positroni; ed n-tuple sul file NTUPLE.Y.OUT, per la descrizione dell'energia persa in ogni rivelatore per ogni storia.

Il file HISTCYL.DAT contiene per ogni istogramma: un identificatore, un titolo, il numero di canali, il valore minimo e massimo della variabile di cui si vuole l'istogramma. Per le n-tuple il file contiene il numero ed i nomi delle variabili. Il file HISTCYL.DAT, contenente le specifiche degli istogrammi, è descritto nella tabella seguente:

HISTCYL.DAT		
SCHEDA	FORMATO	CONTENUTO
1	*	ID numero istogramma NCHX numero canali XMIN valore del canale minimo XMAX valore del canale massimo VALMAX = 0 bit usati da hbook per ogni canale
2	A80	TITLE titolo istogramma - si mette una scheda di tipo 1 ed una di tipo 2 per ogni istogramma monodimensionale; segue una scheda tipo 1 con ID = 0
3	*	ID numero istogramma NCHX numero canali in X XMIN valore del canale X minimo XMAX valore del canale X massimo NCHY numero canali in Y YMIN valore del canale Y minimo YMAX valore del canale Y massimo VALMAX = 0 bit usati da hbook per ogni canale
4	A80	TITLE titolo istogramma - si mette una scheda di tipo 3 ed una di tipo 4 per ogni istogramma bidimensionale; segue una scheda tipo 3 con ID = 0
5	*	ID = 800 identificatore NVAR = 3 numero variabili: - coordinate x, y, z posizione di decadimento del positrone NPRIME
6	A80	TITLE
7	A8	3 schede con nomi coordinate x, y, z
8	*	ID = 801 identificatore NVAR = 16 numero variabili: - numero della storia - numero del rivelatore - x, y, z posizione media dell'energia depositata nel rivelatore - sx, sy, sz dispersione della distribuzione di energia - energia totale depositata nel rivelatore - energia depositata per elettroni, positroni e gamma - numero totale di particelle assorbite dal rivelatore - numero di elettroni, positroni e gamma NPRIME
9	A80	TITLE
10	A8	NVAR = 16 nomi delle variabili, conclude una scheda di tipo 1 con ID = 0

Tabella A3 - Schede del file YAPS.DAT

L'utente modifica questo file solo per ridefinire il numero dei canali, il minimo e massimo valore richiesto ed il titolo degli istogrammi, in funzione del numero di zone, numero di rivelatori, e dimensioni del sistema. Non può modificare gli identificatori degli istogrammi, né eliminarne o aggiungerne. Per le n-tuple, possono essere variati solo i nomi delle variabili. Per eliminare o aggiungere istogrammi o n-tuple occorre modificare le routines: HISTIN\_CYL, che definisce gli istogrammi; AUSGAB, che fa il "booking"; il main, che "chiude" gli istogrammi. Gli istogrammi previsti dal programma sono illustrati nelle seguenti tabelle:

ISTOGRAMMI MONODIMENSIONALI	
IDENTIFICAZIONE	CONTENUTO
101	energia persa nella zona 1
102	energia persa nella zona 2
501	angolo fra direzioni dei fotoni prodotti dal decadimento del positrone

*Tabella A4 - Identificatori degli istogrammi monodimensionali*

ISTOGRAMMI BIDIMENSIONALI	
IDENTIFICAZIONE	CONTENUTO
201	distribuzione x-y dell'energia persa nella zona 1
202	distribuzione x-y dell'energia persa nella zona 2
203	distribuzione x-y dell'energia persa nella zona 3
400	distribuzione x-y dei punti di decadimento dei positroni
401	distribuzione r-z dei punti di decadimento dei positroni
700	distribuzione r-z dell'energia persa
701	distribuzione x-y dell'energia persa

*Tabella A5 - Identificatori degli istogrammi bidimensionali*

N-TUPLE	
IDENTIFICAZIONE	CONTENUTO
800	coordinate x,y,z posizione di decadimento positroni
801	distribuzione energia persa nei rivelatori, 16 variabili: <ul style="list-style-type: none"> <li>- numero della storia</li> <li>- numero del rivelatore</li> <li>- x, y, z posizione media dell'energia depositata nel rivelatore</li> <li>- sx, sy, sz dispersione della distribuzione di energia</li> <li>- energia totale depositata nel rivelatore</li> <li>- energia depositata per elettroni, positroni e gamma</li> <li>- numero totale di particelle assorbite dal rivelatore</li> <li>- numero di elettroni, positroni e gamma</li> </ul>

*Tabella A6 - Identificatori delle N-tuple*

Gli istogrammi sono prodotti sul file HBOOK.OUT (hislun=20), la n-tupla numero 800, coi dati di decadimento dei positroni, sul file NTUPLE.OUT (hislun1=21), la 801, coi dati sui rivelatori, sul file NTUPLE\_Y.OUT (hisluny=22).

## Appendice B

I seguenti listati sono esempi di procedure DCL per la compilazione, il linking e l'esecuzione del codice EGS4CYL. Si noti come nella procedura che effettua il "run" del programma si possano modificare i nomi dei file di input ed output.

```
$! -----
$!  PROCEDURA FOR _CYL_ 2 PER LA COMPILAZIONE DI EGS4CYL
$!  -----
$ wr="write sys $output"
$ wr "      Compilation of user written routines for egs4 _cyl. "
$!
$ if Pl.eqs."DEBUG"
$ then
$   FOR:=FOR/DEBUG/NOOPT
$ endif
$ wr " compiling egs4_cyl_2 ..."
$   for/nolist egs4_cyl_2
$ wr " compiling init_cyl_2 ..."
$   for/nolist init_cyl_2
$ wr " compiling histin_cyl_2 ..."
$   for/nolist histin_cyl_2
$ wr " compiling source_cyl_2 ..."
$   for/nolist source_cyl_2
$ wr " compiling howfar_cyl_2a ..."
$   for/nolist howfar_cyl_2a
$ wr " compiling ausgabe_cyl_2 ..."
$   for/nolist ausgabe_cyl_2
$ wr " compiling statend_cyl_2 ..."
$   for/nolist statend_cyl_2
$ wr "      all including: comm_cyltot_2.for"
$ wr "                      comm_yaps.for"
$ wr " compiling guarda_2.for"
$ !wr "      including: pg_comm.for"
$ !wr "                      comm_cyl_2.for"
$ wr "      now including: comm_cyl_2.for"
$ wr "                      comm_yaps.for"
$ wr "                      petgraf.inc"
$   for/nolist guarda_2
$ wr " compiling yaps.for ..."
$   for/nolist yaps
$ wr " compiling howyaps.for .."
$   for/nolist howyaps
$ wr "      including: comm _yaps.for"
$ wr ""
$ wr " compilation ended : "
$ wr "      files not re-compiled: interfaces to highz : "
$ wr "      petgraf.for , including: petgraf.inc,petgraf_com.inc"
$ wr "      petgraf_data.inc"
$ wr "      petgraf_higz.inc"
$ wr "      egs4funl.for : egs4 routines"
$ wr ""
$ dir *.obj
```

*Procedura per la compilazione di EGS4CYL*

```

$! -----
$!  PROCEDURA LINK_CYL_2 PER IL LINK DI EGS4CYL
$!  -----
$ IF P1 .egs. "DEBUG" then LINK:=LINK/DEBUG
$ wr:=write sys $output
$ wr "      linking egs4 routines"
$!
$! Definizioni di uso generale, che possono essere anche effettuate altrove
$! define egs4f $disk sys $sysdevice:[galli.egs4]
$! define pgslib sys $sysdevice:[galli.pg]petsub.obj ! interface to higz
$! define pglib sys $sysdevice:[galli.pg]petgra.olb ! some higz routines
$! define lib $
$!         cern $library:packlib/lib,-
$!         cern $library:genlib/lib,-
$!         cern $library:kernlib/lib
$! define gksbnd sys $common:[syslib]gksforbnd.olb ! DecGKS fortran interfaces
$!
$! -----
$! altro esempio di definizioni di uso generale:
$!
$!         per il link di egs4_cyl_2 :
$! define pgslib petsub.obj ! interfacce ad higz
$! define pglib cern:[pro.lib]graflib.olb ! routine grafiche cern
$!         ! dec gks fortran interfaces
$! define gksbnd sys $common:[syslib]gksforbnd.olb
$! define gksbnd cern:[pro.lib]GRAFGKS.OLB
$! define materiali mat3mev.dat ! materiali da pegs4
$! -----
$!
$!
$ write sys $output " linking all ..."
$ cernlib mathlib,packlib,graflib/x11,packlib,mathlib
$ link/nomap egs4_cyl_2,init_cyl_2,yaps,-
$         histin_cyl_2,source_cyl_2,-
$         guarda_2,-
$         howfar_cyl_2a,howyaps,ausgabe_cyl_2,-
$         statend_cyl_2,-
$         egs4fun1,-
$         petgraf.obj, -
$         cern $mgr:gethostname, -
$         'lib $'
$!
$ write sys $output " done ..."
$ dir *.exe
$ exit
$!
$! ----- altro esempio di comando link:
$ link/nomap egs4_cyl_2,init_cyl_2,yaps,-
$         histin_cyl_2,source_cyl_2,-
$         guarda_2,-
$         howfar_cyl_2a,howyaps,ausgabe_cyl_2,-
$         statend_cyl_2,-
$         egs4fun1,-
$         pgslib,pglib/lib, -
$         gksbnd/lib, -
$         'lib $'
$ write sys $output " done ..."
$ dir *.exe

```

*Procedura per il link di EGS4CYL*

```

$! -----
$! PROCEDURA RUN_CYL_2_Y PER IL RUN DI EGS4CYL
$! -----
$!      caso con 6 rivelatori di 4.2*4.2*2cm attorno a
$!      cilindretto di aria di raggio r=2cm, circondato
$!      da spessore di alluminio di 0.5cm
$!
$ wr:=write sys $output
$! wr " running egs4cyl IF P2.EQS. PATHS prints particles path"
$ wr "                               IF P1.eqs. "" "" nodebug"
$!
$ if P1 .eqs. ""
$ then
$   P1="NODEBUG"
$ else
$   P1="DEBUG"
$ endif
$ write sys $output " now running egs4_cyl_2 ..."
$ define materiali mat3mev.dat
$!
$!
$ assign/user   cyl_2_y.dat           for005      ! terminal input
$ assign/user   yap5_y.dat           for006      ! terminal output
$ assign/user   hatch_y.out          for007      ! geometry input
$ assign/user   hatch_y.out          for013      ! geometria rivelatori
$ assign/user   hist_cyl_2_y.dat     for008      ! egs4 mateiral out list
$ assign/user   hist_cyl_2_y.dat     for009      ! histogram input
$ assign/user   cyl_y.out            for010      ! output
$!
$ assign/user   materiali             for011      ! ps metafile for plots
$ assign/user   hbook_y.out          for012      ! mat library
$ assign/user   hbook_y.out          for020      ! hbook output
$ assign/user   ntuple_y.out         for021      ! hbook output
$ assign/user   ntuple_y.out         for022      ! hbook output
$! if P2 .eqs. "PATHS"
$! then
$ assign/user   particles.out         for050      ! particl `es path (for test)
$! else
$! assign/user   nl:                  for050      ! no particl `es path
$! endif
$ assign/user sys $command sys $input ! user input from terminal
$ run/'P1' egs4_cyl_2
$ write sys $output " run ended ....."
$ write sys $output " printed output in:   CYL.OUT "
$ write sys $output " hatch   output in:   HATCH.OUT "
$ write sys $output " HBOOK  output in:   HBOOK.OUT "
$ write sys $output " Ntuple  output in:   ntuple.OUT "
$ if P2.eqs."PATHS" then wr "particles paths in: PARTICLES.OUT"
$ exit

```

*Procedura per l'esecuzione di EGS4CYL*

## Appendice C

I listati che seguono sono una stampa dei file di input utilizzati per la simulazione del semplice apparato illustrato nella sezione 6. Infine é riportato il sommario dei risultati forniti dal codice.

```
40000 1 1 1.00 ! N.storie,carica,regione,mev sorgnte
90000 1 -122545931 ! stampe ogni 90000, user seed flag, e seed
5 ! numero materiali
WATER-3MEV-20KEV-20KEV
ALUM-3MEV-20KEV-20KEV ! nomi dei materiali coerentemente
AIR-3MEV-20KEV-20KEV ! con quanto specificato in MAT.DAT
YAPO3-3MEV-20KEV-20KEV
LEAD-3MEV-20KEV-20KEV
1 0.002 0.002
2 0.002 0.002 ! n.materiale, E inf. gamma e positroni
3 0.002 0.002
4 0.002 0.002
5 0.002 0.002
2 2.1 ! num cilindri concentrici, semialtezza
2. ! spessore , materiale cilindro (aria)
0.5 2 ! spessore , materiale (alluminio)
1 0 1 1 0 0 ! output flags
```

*File CYL.DAT*

```

6 0. 0. 2. 4.2 4.2 3 4 !num.rivelatori, r,z,dx,dy,dz,mezzo, mezzo riv.
0.0 1.0 !angolo collimatore, flag uscite dal rivelatore

```

*File YAPS.DAT*

```

101      60      -3.      3.      0. Zona 1 : Aria r=2cm : Distribuzione energia ceduta lungo l' asse z
102      60      -3.      3.      0. Zona 2 : Al r=0.5cm : Distribuzione energia ceduta lungo l' asse z
103      60      -3.      3.      0. Zona 3 : aria : Distribuzione energia ceduta lungo l' asse z
104      40      -2.      2.      0. Zona 4 : YAP 1 : Distribuzione profondita energia ceduta
105      40      -2.      2.      0. Zona 5 : YAP 2 : Distribuzione profondita energia ceduta
106      40      -2.      2.      0. Zona 6 : YAP 3 : Distribuzione profondita energia ceduta
107      40      -2.      2.      0. Zona 7 : YAP 4 : Distribuzione profondita energia ceduta
108      40      -2.      2.      0. Zona 8 : YAP 5 : Distribuzione profondita energia ceduta
109      40      -2.      2.      0. Zona 9 : YAP 6 : Distribuzione profondita energia ceduta
110      60      -3.      3.      0. Zona 10 : fughe dal sistema : Distribuzione lungo asse z
111      100     -5.      5.      0. Zona 11 : assorbimenti da parte dei collimatori distr. asse z
112      100     -5.      5.      0. Zona 12 : particelle uscite dai rivelatori distr. asse z
501      72      0.      6.28  0. Distribuzione angolo decadimento positroni
0        0        0.      0.      0.
201 100 -10. 10. 100 -10. 10. 0.
Zona 1 : aria r=0.5cm : Distribuzione energia ceduta nel piano x-y
202      100     -10. 10. 100 -10. 10. 0.
Zona 2 : Al r=0.5cm : Distribuzione energia ceduta nel piano x-y
203      100     -10. 10. 100 -10. 10. 0.
Zona 3 : Aria Distribuzione energia ceduta nel piano x-y
204      70      -2.1 2.1 70 -2.1 2.1 0.
Zona 4 : YAP 1 Distribuzione energia ceduta nel piano del yap
205      70      -2.1 2.1 70 -2.1 2.1 0.
Zona 5 : YAP 2 Distribuzione energia ceduta nel piano del yap
206      70      -2.1 2.1 70 -2.1 2.1 0.
Zona 6 : YAP 3 Distribuzione energia ceduta nel piano del yap
207      70      -2.1 2.1 70 -2.1 2.1 0.
Zona 7 : YAP 4 Distribuzione energia ceduta nel piano del yap
208      70      -2.1 2.1 70 -2.1 2.1 0.
Zona 8 : YAP 5 Distribuzione energia ceduta nel piano del yap
209      70      -2.1 2.1 70 -2.1 2.1 0.
Zona 9 : YAP 6 Distribuzione energia ceduta nel piano del yap
210      100     -10 10 100 -10 10 0.
Zona 10 : fughe dal sistema : distribuzione in x-y
211      100     -10. 10. 100 -10. 10. 0.
Zona 11 : assorbimento dei collimatori distribuzione x-y
212      100     -10. 10. 100 -10. 10. 0.
Zona 12 : particelle uscite dai rivelatori, distr x-y
700      60      0.      6.      60 -3. 3. 0.
DISTRIBUZIONE R-Z , PESATA SU ENERGIA --
701      120     -6.      6.      120 -6. 6. 0.
DISTRIBUZIONE x-y , PESATA SU ENERGIA --
400      120     -6.      6.      120 -6. 6. 0.
DISTRIBUZIONE X-Y PUNTI DECADIMENTO POSITRONE
401      60      0.      6.      60 -3. 3. 0.
DISTRIBUZIONE R-Z PUNTI DECADIMENTO PPOSITRONE
0        0        0.      0.      0 0. 0. 0.
800      3        10
x,y,z punti decadimento del positrone
x-decay
y-decay
z-decay
801      16      10
energia persa nei rivelatori a ogni evento
shower
rivelatore
xmedio
ymedio
zmedio
sigmax
sigmay
sigmaz
etot
etot-positroni
etot-elettroni
etot-gamma
num-part
num-positroni
num-elettroni
num-gamma
0        0        0        0        0
FINE DEGLI HISTOGRAMMI

```

*File Hist.dat*

## SOMMARIO DEI RISULTATI

ENERGIA TOTALE POSITRONI	:=	60460.15	MeV
ENERGIA CINETICA TOTALE	:=	40000.00	MeV
ENERGIA TOTALE DEPOSITATA	:=	80880.27	MeV
ENERGIA INIZIALE/DEPOSITATA	:=	1.337745	MeV
NUMERO POSITRONI	:=	40000	

REGIONE	PERCENTUALE	MeV
1	0.21601	174.71
2	36.46807	29495.47
3	0.00076	0.62
4	0.84370	682.39
5	0.88416	715.11
6	0.84897	686.65
7	0.82928	670.72
8	0.85157	688.75
9	0.86973	703.44
10	46.25805	37413.64
11	0.00000	0.00
12	11.92968	9648.75

### FRAZIONE DI ENERGIA DEPOSITATA IN OGNI REGIONE

REGIONE	TOTALE	e <sup>+</sup>	e <sup>-</sup>	γ
1	0.22E-02	0.19E-02	0.21E-03	0.57E-08
2	0.36E+00	0.28E+00	0.83E-01	0.64E-04
3	0.77E-05	0.00E+00	0.77E-05	0.00E+00
4	0.84E-02	0.00E+00	0.83E-02	0.99E-04
5	0.88E-02	0.00E+00	0.87E-02	0.10E-03
6	0.85E-02	0.00E+00	0.84E-02	0.97E-04
7	0.83E-02	0.00E+00	0.82E-02	0.96E-04
8	0.85E-02	0.00E+00	0.84E-02	0.98E-04
9	0.87E-02	0.00E+00	0.86E-02	0.99E-04
10	0.46E+00	0.30E+00	0.34E-02	0.16E+00
11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	0.12E+00	0.00E+00	0.15E-03	0.12E+00

### PARTICELLE ASSORBITE IN OGNI REGIONE

REGIONE	TOTALE	e <sup>+</sup>	e <sup>-</sup>	γ
1	747	0	746	1
2	199656	26863	169467	3326
3	1	0	1	0
4	8631	0	7538	1093
5	8991	0	7886	1105
6	8678	0	7594	1084
7	8556	0	7484	1072
8	8627	0	7533	1094
9	8805	0	7705	1100
10	41997	12150	1677	28170
11	0	0	0	0
12	23312	0	56	23256

## **Bibliografia**

- 1 R.Pani et al.  
"Multi-crystal YAP:Ce Detector System for Position Sensitive Measurements"  
Third London Conference on Position-sensitive Detector, Brunel 6-10 sept. 93
  
- 2 Walter R.Nelson, Hideo Hirayama and David W.O. Rogers  
The EGS4 Code System  
SLAC Report-265 - December 1985

**Edito dall'Enea  
Funzione Centrale Relazioni Esterne  
V. le Regina Margherita, 125 - 00198 Roma  
Finito di stampare nel mese di dicembre 1995  
presso il Tecnografico**