CRN STRASBOURG

CRN 92-22

MULTI-DIMENSIONAL ANALYSIS OF HIGH RESOLUTION γ-RAY DATA

S. Flibotte, U.J. Hüttmeier, G. de France, B. Haas,
P. Romain, Ch. Theisen, J.P. Vivien and J. Zen
Centre de Recherches Nucléaires,
IN2P3-CNRS/Université Louis Pasteur,
F-67037 Strasbourg Cedex, France

Accepted for publication by Nucl. Instr. and Meth. A

CENTRE DE RECHERCHES NUCLEAIRES

STRASBOURG

IN2P3
CNRS

UNIVERSITE
LOUIS PASTEUR

MULTI-DIMENSIONAL ANALYSIS OF HIGH RESOLUTION γ-RAY DATA

S. Flibotte, U.J. Hüttmeier, G. de France, B. Haas,
P. Romain, Ch. Theisen, J.P. Vivien and J. Zen
Centre de Recherches Nucléaires,
IN2P3-CNRS/Université Louis Pasteur,
F-67037 Strasbourg Cedex, France

Abstract

Algorithms are developed to analyze high-fold γ-ray coincidences. Performances of the programs have been tested in 3-, 4- and 5-dimensions using events generated with a Monte Carlo simulation.

1 Introduction

In the last decade, γ-ray spectrometers based on arrays of many Compton-suppressed high purity Ge detectors have proved to be powerful instruments to study nuclear structure at very high spin. Until now they have been designed principally to measure high resolution 2-fold energy coincidences. In order to answer many exciting questions raised by experiments performed with these spectrometers, a new generation of multidetectors capable of measuring higher-fold coincidences with a large efficiency is presently under construction (EUROGAM, GASP, GAMMASPHERE). The future experimental progress of our understanding of nuclear structure depends critically on our ability to analyze the coincidence data in a multi-dimensional space and to resolve small photopeaks of interest from the generally large background. The development of programs to process such high-fold events is still in its infancy and only the 3-fold case has been treated so far [1-6]. As a contribution to the software development associated with the EUROGAM spectrometer, we have written and tested the performances of computer codes designed to select multi-dimensional gates from 3-, 4- and 5-fold coincidence databases. The tests were performed on events generated with a Monte Carlo simulation and also on real experimental triple data recorded with the 8π spectrometer [7] and with a preliminary version of the EUROGAM array.

2 Methods

Our principal goal was to develop appropriate software to analyze large sets of f-fold coincidence events by setting (f-1)-dimensional gates to obtain the resulting spectra as fast as possible. The basic concepts used in the computer codes will be presented in this section.

2.1 Database structure

Obviously, the now classical technique developed to analyze γ-γ coincidences — the slicing of a 2-dimensional 4096×4096 channel matrix — cannot be extended to higher-fold data. For example, if we assume that the content of each channel is represented by one byte, the storage of a 4-dimensional histogram with 4096 channels per axis would require a total of 256 Terabytes (Tb) of memory, while the capacity of the largest commercially available hard disks is of the order of a few Gigabytes (Gb). Furthermore, such a matrix would be mostly empty since a typical high-spin experiment performed with a spectrometer of the new generation will produce of the order of 10^9-10^{10} 4-fold events resulting in a density of $\rho \simeq 10^{-5}$ counts/channel. This implies that the memory requirement can be reduced by storing the data in the form of lists of events. In list mode, the memory needed to save the complete database is proportional to the total number of events and also to the number of bytes used to store each individual event. Therefore, the hard disk space requirement can be reduced by rejecting events associated with the Compton background [2, 8] or by changing the energy dispersion with a linear or nonlinear compression procedure [9].

Simplifications occur on the database structure, and consequently on the algorithms of the programs, when the analysis is restricted to one particular fold f at a time. In practice, a physical event which has triggered m Ge detectors ($m \geq f$) will be interpreted as $\binom{m}{f}$ f-fold coincidence events. The minimal number of bits required to store each event is reduced if the f-dimensional space is divided and subdivided into smaller subspaces [4]. All events located in a given subspace share a common f-dimensional energy offset associated with the position of the subspace and only the energy differences relative to this offset have to be stored. Furthermore, this data structure increases the speed of the slicing process since only a certain number of subspaces need to be treated for a particular gate. For many high-spin experiments, the position information on the Ge detectors which have fired can be neglected in the analysis. In order to decrease the total number of subspaces, the f energy parameters $\{E_i\}$ representing an event can then be stored in numerical order, e.g.:

$$E_i \leq E_j, \ \forall j > i \,. \tag{1}$$

Because we lose the information concerning the hit pattern, the software discussed in this paper will not be applicable for data analysis where the angular information of each Ge counter is important such as, for example, in DSAM (Doppler-Shift Attenuation Method) experiments [10].

Fig. 1 shows a graphical representation of the database structure for the 2-fold case. In this example, the total dispersion on each axis is divided into four equal sections to obtain ten 2-dimensional subspaces. The term "file" will be used to denote these subspaces since they correspond to UNIX files in our implementation. In the general f-fold case, if the total number of channels per axis is C_T and the width of each file is C_F, then the total number of files N_F is given by the combinatorial expression:

$$N_F = \binom{\frac{C_T}{C_F} + f - 1}{f} \,. \tag{2}$$

Each file is further subdivided into subspaces called "lists" where the events are actually stored. If the width of the lists in channels is represented by C_L, then the number of lists per file is:

$$N_L = (\frac{C_F}{C_L})^f \,. \tag{3}$$

The parameters C_T, C_F and C_L are chosen in such a way that the ratios C_T/C_F and C_F/C_L are integer. The total number of lists N_T in the database is given by the product of N_F and N_L:

$$N_T = N_F \times N_L \,. \tag{4}$$

It is important to know this quantity since the program has to create a vector to save the number of events written in each list. For efficiency reasons, this vector should be kept in DRAM (Direct Random Access Memory). This means that the upper limit of N_T is related to the amount of physical memory available in the computer used.

Each list is uniquely identifiable by its list number I_L and its file number I_F. An event $\{E_i\}$ is assigned to its specific list using the relations:

$$I_F = \sum_{i=1, n_i^F \neq 0}^{f} \binom{n_i^F + i - 1}{i} \tag{5}$$

and

$$I_L = \sum_{i=1}^{f} (\frac{C_F}{C_L})^{f-i} n_i^L \,, \tag{6}$$

where

$$n_i^F = E_i/C_F \tag{7}$$

and

$$n_i^L = (E_i \, mod \, C_F)/C_L \,. \tag{8}$$

In the last two equations, the symbols / and *mod* represent the integer division and modulo operators, respectively. Since $\{n_i^F\}$ and $\{n_i^L\}$ are common to all the events in a given list, only the energy differences $\{E_i'\}$ with respect to the list energy offsets have to be stored inside the list:

$$E_i' = E_i - C_F n_i^F - C_L n_i^L \,, \tag{9}$$

which can be expressed more concisely by the equation:

$$E_i' = (E_i \, mod \, C_F) \, mod \, C_L \,. \tag{10}$$

Therefore, the number of bits b required to store each f-fold event is:

$$b = f \log_2 C_L \,. \tag{11}$$

Besides reducing the memory requirement, the adopted database structure permits a fast slicing algorithm. Assuming that a given gate does not overlap any list boundaries, only C_T/C_L lists will have to be read from the disk and processed. An example of a slicing trajectory is indicated by the black area in Fig. 1, while the relevant lists from the relevant files are indicated by the hatched area. In this particular example, the window test is performed on the parameter E_2 for the horizontal portion of the slicing trajectory and on the parameter E_1 for the vertical one. For the list located on the diagonal, both parameters are tested. For a given event, if the tested parameter (corrected for the energy offsets) falls inside the energy window of the gate, the channel associated with the other energy parameter is incremented in the projection histogram. With our database structure, the slicing technique can be easily generalized to perform (f-1)-dimensional gates on f-fold coincidences. The slicing trajectory is calculated with the help of equations 5 to 10, one particular energy parameter being free to vary in a given slicing direction.

2.2 Monte Carlo simulation of f-fold coincidences

Presently, there is no existing experimental data set large enough to provide high statistics test of our analysis programs for a fold larger than three. To get around this situation, f-fold events were generated using a Monte Carlo technique. The Monte Carlo simulation has the advantage of giving a better control over the data set characteristics, which facilitates programming error tracking and the comparison of the software performance between different folds. We have imposed two important constraints on the simulation design: it had to be as little as possible CPU time consuming and the generated events had to be distributed in the f-dimensional space more or less like the distribution expected for an experimental data set.

The generation of a spectrum following the distribution $s(E)$ in one-dimension is a straightforward task. Defining by $S(E)$ the cumulative distribution function of $s(E)$

$$S(e) = \frac{1}{A}\int_0^e s(E)dE \tag{12}$$

where A is a normalization factor

$$A = \int_0^{C_T} s(E)dE \tag{13}$$

each one-dimensional event can be generated using the relation

$$E = S^{-1}(u) \tag{14}$$

where u is a random number drawn between 0 and 1. The simplest way to simulate f-fold events is to choose a set of f such γ-ray energies E. This means that we assume that each γ ray in the generating spectrum $s(E)$ is emitted in coincidence with all the others. To be more realistic, events including photopeak-photopeak auto-coincidences have been rejected in order to simulate the well-known valley structure along the diagonals. For each f-fold event, the association of a particular energy E_i with the photopeak or the background have been chosen randomly with a relative weight extracted from a smooth background curve drawn under the peaks present in the $s(E)$ distribution. To be able to study the change in resolving power from one fold to another, two different generators $s(E)$ were used, one generator being associated with a contamination. For a given event, its generating spectrum was randomly selected with a relative probability fixed by the user.

3 Implementation

The implementation of the algorithms already presented will be discussed in this section. Computer programs were developed to analyze 3-, 4- and 5-fold coincidence events.

3.1 Three phase procedure

The problem of multi-dimensional gating explained in Section 2 can be artificially divided into three phases. In our implementation, an independent computer program corresponds to each different phase.

The first phase program has the responsibility of reading the data recorded by the multidetector and extracting only the events corresponding to the requested fold, the events of higher-fold being "unfolded" to the studied fold. Other tests and processes such as a gain alignment can also be performed on the raw data in this initial step. In the present work, the reading function was replaced by our Monte Carlo simulation of f-fold coincidences. The selected events are then stored on hard disk using the file structure described earlier. At this stage, the data are already stored in a compressed format, each energy parameter typically requiring one byte.

In the second phase, each individual file is processed to create the list structure. A special file containing the number of events stored in each list is also created and saved for the slicing program.

The third step corresponds to the slicing procedure. The program reads a gate file created by the user and extracts the resulting spectrum from the database. This spectrum corresponds to the sum of all combinations of the multi-dimensional gates obtained from the energy windows present in the gate file. This third phase can be repeated as many times as desired with different gate files.

The total number of channels per axis and the width of each file and list are selected by the user. The optimal set of parameters depends on the total number of events to be processed and also on the available hardware configuration. The values chosen in the present work are presented in Table 1 for 3-, 4- and 5-fold data analysis. Real experimental events are usually recorded on magnetic tape using two bytes per energy parameter. A data compression factor of up to four is achieved with the adopted set of parameters given in Table 1.

3.2 Computer hardware, compatibility and utility programs

All the programs were written in C++ [11], an objet-oriented programming language. Programs were developed and tested on a single-processor Sun SPARC-2 with a performance of about 28 MIPS and 6.2 Mflops. The system was equipped with a 1.2 Gb Seagate hard disk with a mean access time of 15 ms and a SCSI-2 interface with a tested data transfer rate of about 4 Mb/s for reading and 500 kb/s for writing operations.

In the computer codes, input/ouput operations are performed with functions from the standard Sun C library [12]. The resulting spectra are stored in a format compatible with GPSI [13], the spectral analysis program used in Strasbourg. Besides the Monte Carlo generated events, the software presently handles data written in 8π [14] and in Daresbury formats, and is easily modified for other formats.

Some utility programs were also written to manipulate the database. For example, one of them can combine two different data sets, which is particularly useful because it permits filling up a disk with completely compressed events.

4 Results

The two generating spectra used in the Monte Carlo simulation are shown in Fig. 2. They were acquired in the high-spin experiment performed with the 8π spectrometer which permitted the discovery of excited superdeformed (SD) rotational bands populated in the ^{149}Gd nucleus [7]. In the simulation, the SD generator was selected for 5% of the events. A total of 3×10^8 events were generated in 2-, 3-, 4- and 5-folds. The software performance was tested by gating on SD transitions to study our ability to isolate an SD spectrum from the large background simulated by the normal-deformed generator. In the 2-fold case, the data analysis was performed with a classical slicing technique.

Due to the limited capacity of the hard disk used, it was not possible to generate 3×10^8 events in a single program execution. Two data sets of 1.5×10^8 events each were then generated and compressed for each fold under consideration. In the 3- and 4-fold cases, it was possible to combine the two different data sets to form a large database containing 3×10^8 events for each fold. The CPU time required to execute the first two phases described in Section 3.1 are presented in Table 2 for a data set comprising 1.5×10^8 events. The phase I CPU time is proportional to the number of generated events. In an analysis of real experimental data, this time is dominated by the data transfer rate of the Exabyte tape drive. For example, extrapolating from the time required to process an existing 8π data set containing 5×10^7 3-fold events [7], the reading of 1.5×10^8 3-fold events from Exabyte at a rate of 250 kb/s would require ≈400 min., while the phase I procedure would need only ≈200 min. of CPU time. The time required to execute the second phase increases less rapidly with the number of events. For example, the phase II CPU time for 4-fold data is roughlv proportional to the cube root of the number of events processed, at least in the studied range (5 to 150 million events). The CPU time of the second phase for 5-fold data reported in Table 2 is relatively large compared to the other folds. This is explained by the very large number of events per file and the physical memory (DRAM) limitation. In this case, the algorithm was slightly modified to process each file in four steps, which explains the factor of four between the reported time and the value expected from the systematic trend. However, this is a minor issue since the second phase has to be performed only once for each data set.

Superdeformed band spectra were obtained by summing all combinations of multi-dimensional gates placed on the 17 SD yrast transitions ranging from 664 to 1501 keV. The average width of the selected energy windows was equal to 6 keV and the same set of energy windows was used for every studied fold. Fig. 3 shows the resulting SD spectra and the program statistics are presented in Table 3. The gain in resolving power by using higher-folds is clearly evident. This gain is expected to be even larger for experimental data since the contaminations will not always be correlated like in our simulation. The slicing CPU time does not strongly depend on the fold and it stays within very acceptable values. This time depends on the specific gate file used and also on the number of events present in the database. The slicing of a multi-dimensional gate involving an energy window overlapping list boundaries takes a longer time since more lists have to be read and processed. For the same number of events in the database, the number of counts in a specific photopeak in the final SD spectrum decreases by a factor of ~6 by going from a given fold to the next higher one. It may be very important to know this factor since it could help to evaluate the beam time required to perform some specific high-spin experiments with the new generation of multidetectors. Concerning the absolute number of counts in the photopeak at 1276 keV, one should recall that the values quoted in Table 3 were obtained when the SD generator was selected for 5% of the events. This is realistic for SD bands populated in the $A \approx 130$ and 190 mass regions.

5 Summary

A new generation of high resolution γ-ray spectrometers capable of recording high-fold coincidence events with a large efficiency will soon be available. In the present work, we have developed algorithms to analyze such events in a multi-dimensional space. The algorithms were implemented for 3-, 4- and 5-fold coincidences. The performances of the programs were tested with large data sets generated with a Monte Carlo simulation. The slicing process requires typically less than one second of CPU time per gate on a Sun SPARC-2. The needed hard disk storage space was reduced to four bits per energy parameter for 3- and 4-fold events and down to 6 bits in the 5-fold case.

References

[1] K. Mossberg, P.O. Forsgren, L. Hildingsson, W. Klamra and Th. Lindblad, Nucl. Instr. and Meth. **A278** (1989) 755.

[2] J. Junikka and A. Lampinen, Nucl. Instr. and Meth. **A292** (1990) 677.

[3] D.C. Radford, private communication.

[4] R.M. Koekoek, Research Reports, University of Jyväskylä, Finland (1990).

[5] T. Lönnroth, Contributed Paper, Fourth Int. Conf. on Nucleus-Nucleus Collisions, Kanazawa, Japan, June 10-14 (1991).

[6] W. Urban, EUROGAM Data Analysis Meeting, C.S.N.S.M. Orsay, Oct. 18, 1991.

[7] B. Haas *et al.*, Phys. Rev. C **42** (1990) R1817.

[8] D.J.G. Love, Nucl. Instr. and Meth. **A302** (1991) 143.

[9] G. Szekely, Th. Lindblad, L. Hildingsson and W. Klamra, Nucl. Instr. and Meth. **A292** (1990) 431.

[10] T.K. Alexander and J.S. Forster, Adv. Nucl. Phys. **10** (1979) 197.

[11] B. Stroustrup, The C++ Programming Language, Addison-Wesley, Reading, Massachusetts, (1986).

[12] SunOS Reference Manual, Sun Microsystems Inc., Mountain View, California, (1988).

[13] J. Zen and P. Wagner, unpublished.

[14] J.P. Martin *et al.*, Nucl. Instr. and Meth. **A257** (1987) 301.

Table 1. Total number of channels per axis (C_T) and width in channels of each file (C_F) and each list (C_L) used to analyze f-fold events. The resulting number of files (N_F) and lists per file (N_L) are presented together with the total number of lists (N_T) in the database. The number of bits (b) required to store each event is also shown.

f	C_T	C_F	C_L	N_F	N_L	N_T	b
3	4096	128	16	5984	512	3.1×10^6	12
4	2048	128	16	3876	4096	15.9×10^6	16
5	2048	256	64	792	1024	0.8×10^6	30[†]

Table 2. CPU times required to generate and compress 1.5×10^8 events. The reported CPU times include both the user time and the system time used [12].

Fold	Phase I (min.)	Phase II (min.)	Total (min.)
3	369	44	413
4	481	40	521
5	602	164[‡]	766

[†] an integer word of 32 bits was actually used in the program implementation.

[‡] see Section 4 text for the explanation of this large value.

Table 3. Slicing program statistics as a function of the fold for two different sets of superdeformed energy windows ranging from 664 to 1501 keV. The very contaminated 712 and 1167 keV transitions were removed in the case of the reduced set of gates. The number of selected transitions and the total number of gate combinations are presented together with the CPU time (user + system [12]) required for the slicing procedure. Databases containing 3×10^8 events were processed except for the 5-fold case where two data sets of 1.5×10^8 events each were used. The total number of counts in the resulting spectra is shown with the number of counts in the E_γ = 1276 keV photopeak.

fold	number of		CPU time (sec)		number of counts	
	transitions	gates	total	per gate	spectrum	γ1276
2 †	15	15	—	—	1.42×10^7	17 490
	17	17	—	—	1.80×10^7	21 919
3	15	105	252	2.40	2.14×10^6	4545
	17	136	386	2.84	3.47×10^6	7359
4	15	455	104	0.23	1.72×10^5	502
	17	680	165	0.24	3.52×10^5	1054
5	15	1365	2×239	2×0.18	1.62×10^4	74
	17	2380	2×456	2×0.19	4.14×10^4	173

† a standard slicing technique has been used for the analysis of 2-fold data.

Figure captions

Fig. 1: Schematic representation of the database structure for a 2-fold coincidence analysis. In this example, the total space is divided and subdivided by a factor of four on each axis. One possible slicing trajectory is indicated in black (see Section 2.1 text).

Fig. 2: Generating spectra used in the Monte Carlo simulation. They were obtained in an experiment performed with the 8π spectrometer [7] In the simulation, the superdeformed (SD) generator was selected for 5% of the events.

Fig. 3: Superdeformed band spectra obtained by summing all combinations of gates set on the 17 SD transitions between 664 and 1501 keV. No background has been subtracted. The transitions associated with the contamination (e.g. 944 keV) are clearly suppressed by using higher-fold coincidences, while the SD transitions (e.g. 1276 keV) are strongly enhanced.

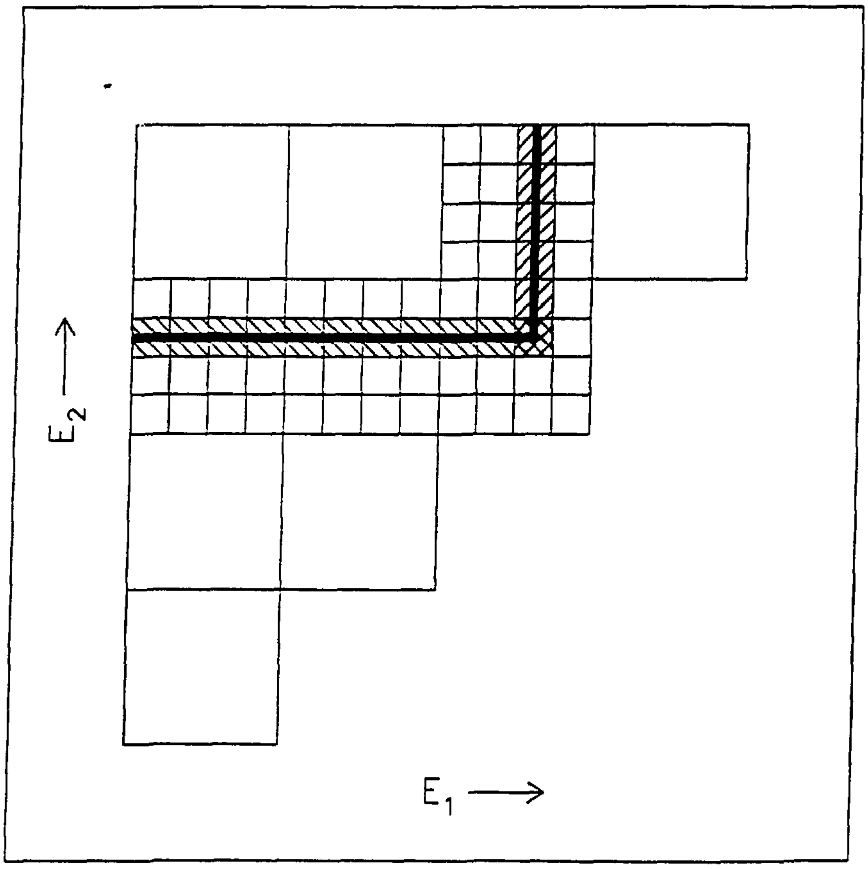
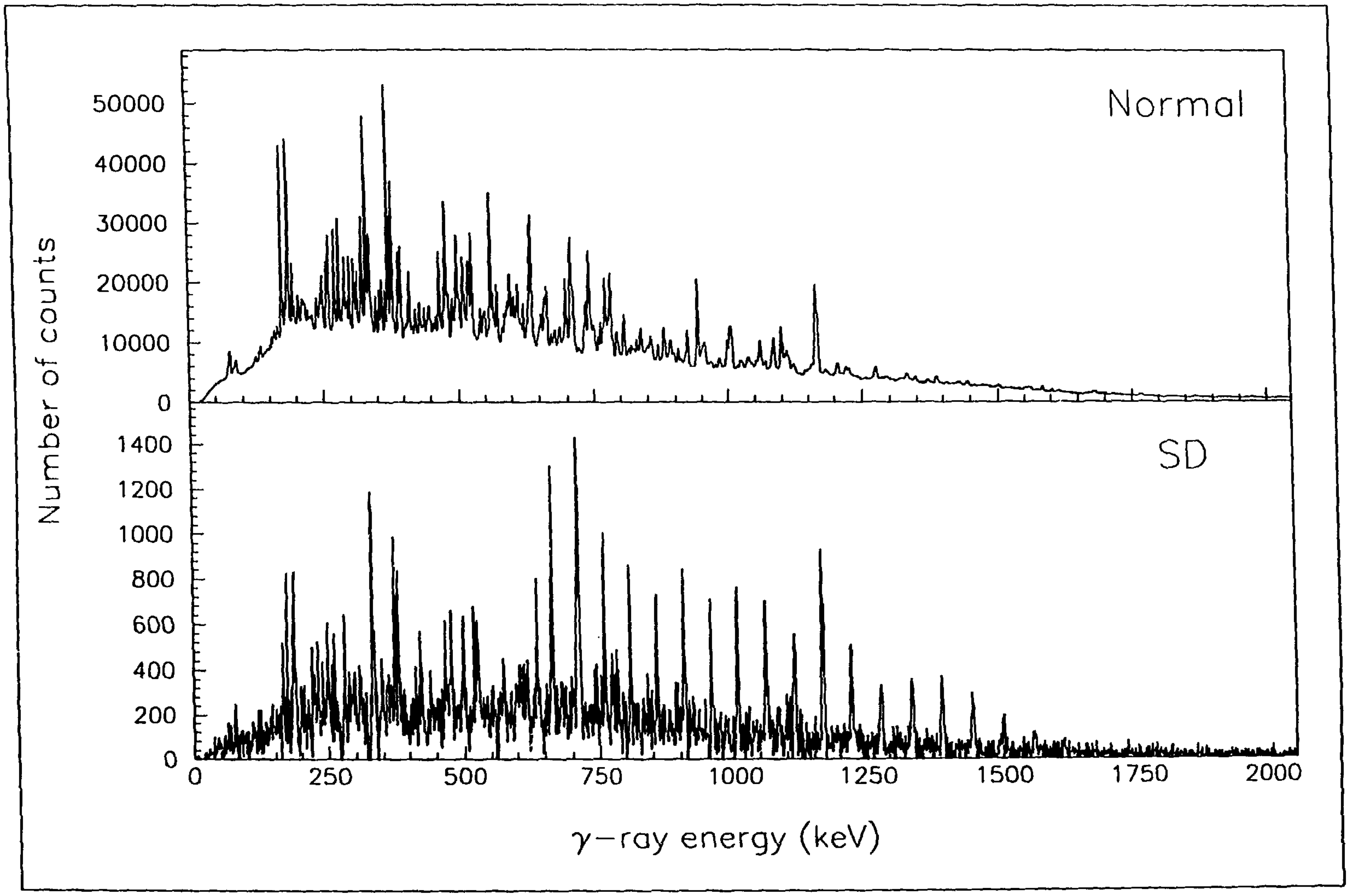


Fig. 1

Fig. 2


Normal
SD
Number of counts
50000
40000
30000
20000
10000
0
1400
1200
1000
800
600
400
200
0
0
250
500
750
1000
1250
1500
1750
2000
γ−ray energy (keV)

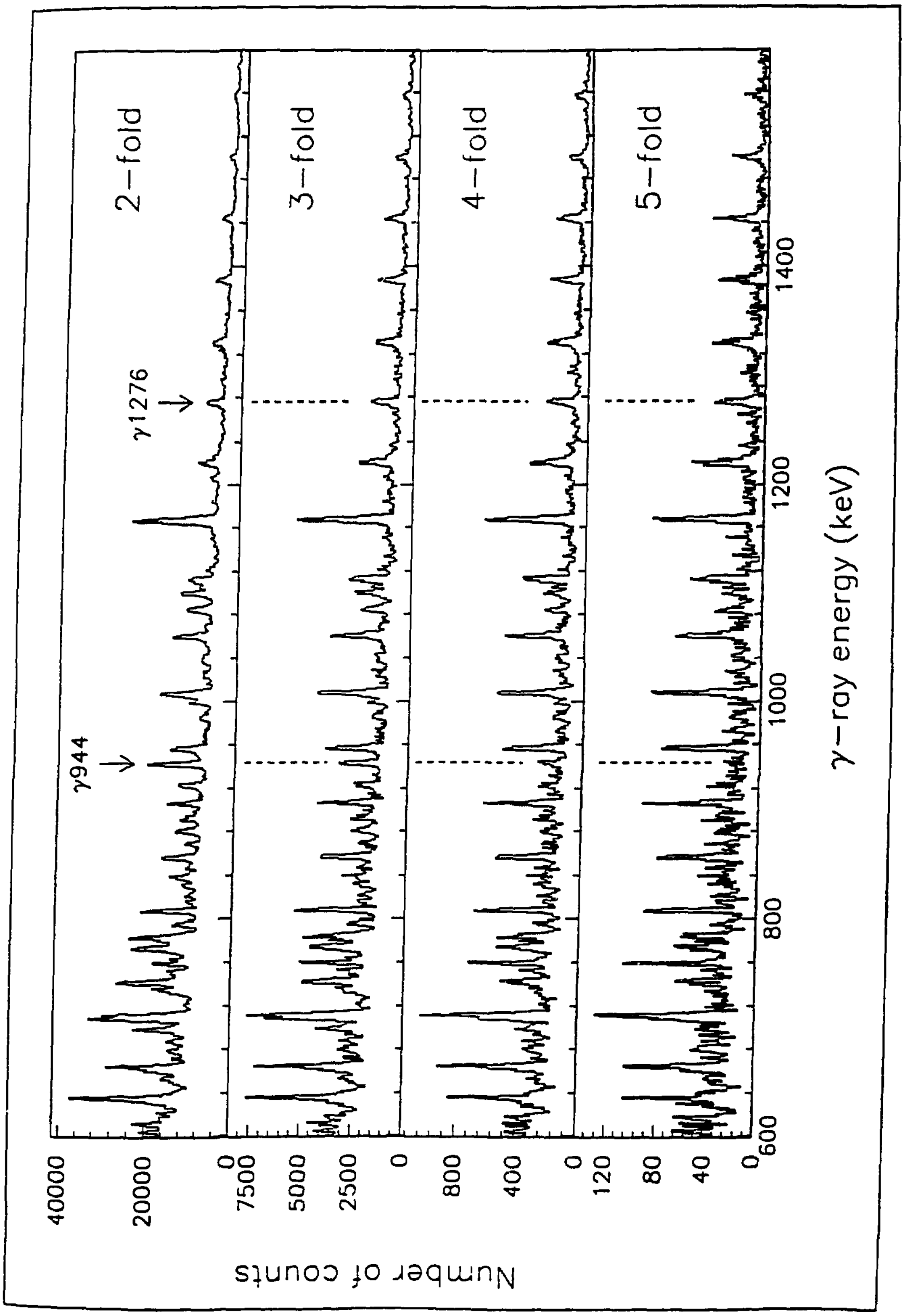

Number of counts
40000
20000
0
7500
5000
2500
0
800
400
0
120
80
40
0
600
800
1000
1200
1400
γ−ray energy (keV)
2−fold
3−fold
4−fold
5−fold
γ944
γ1276

Fig. 3