GSORT PROGRAM

Part of the GASP Data Analysis Program Package

Dino Bazzacco and Calin A. Ur

INFN, Sezione di Padova, Italy

May 13, 1997

TABLE OF CONTENTS

1. INTRODUCTION

The **GSORT** program is part of the Data Analysis Program Package developed at Padova/Legnaro designed to treat data from *GASP/EUROBALL*. It can execute the following operations:

1. projection of raw data;
2. recalibration of the raw data;
3. addback;
4. passing form composite detectors to ID ordered single detectors;
5. tape-to-tape (reduced format);
6. sort data in 1D/2D/3D/4D spectra;
7. sort data in gated 1D/2D/3D/4D spectra ;
8. sort data in multiply gated 1D spectra (cubic or spherical gates).

The sequence of operations to be executed by the program is given in a SETUP file. The SETUP file for sorting *GASP* or *EUROBALL* data is structured according to the different stages of the event analysis, as follows:

1. *FORMAT* Section
   contains the description of the event and defines new parameters to be used during the offline analysis;
2. *DECLARATION* Section
   selection of the events according to some general rules which have to be obeyed by all the events;
3. *ANALYSIS* Section
   contains commands for the effective analysis of the data, as: calibration of the parameters, their ranges and multiplicities; defines the final action to be done during the sort procedure: projecting the data, copying the data or building coincidence cubes and matrices.

The commands specific to each section are presented below. Note that commands belonging to different sections cannot be mixed. Sections has to be specified in the order 1→2→3.

Commands are presented as follows:

1. First row : name
2. Second row: effect
3. Third row : syntax

WARNING !!! Channels and parameters within a detector are counted starting from 0.

2. SUMMARY OF THE SORTING COMMANDS

2.1 FORMAT

- *GASP*
  Define GASP type event format (DEFAULT)

  GASP

- *EUROBALL*
  Define EUROBALL type event format

  EUROBALL

- *GAMMASPHERE*
  Define GAMMASPHERE type event format (only Ge data implemented)

  GAMMASPHERE

- *HEADER*
  Define fixed parameters named 'F' (header of the event)

  HEADER F #pars Res{#pars} [**PLUS #pars+ Res{#pars+}**]

- *DETECTOR*
  Define a detector type named 'D'

  DETECTOR D #detectors #pars Res{#pars} [**PLUS #pars+ Res{#pars+}**]

- *CDETECTOR*
  Define a composite detector type named 'C'

  CDETECTOR C #detectors #segments #pars Res{#pars} [**PLUS #pars+ Res{#pars+}**]

2.2 DECLARATIONS

- *RAWFOLDMIN*
  Define minimum fold to accept event from tape

  RAWFOLDMIN foldmin{for every defined detector}

- *HGATEDEF*
  Define a multiple gate to be applied to a parameter during sort

  HGATEDEF Pn #gates

  Wl Wh {#gates lines}

  HGATEDEF Pn file_with_gates [Individual|Same_for_all]

- *PAIRDEF*
  Define a list of indexed pairs of detectors

  PAIRDEF file_with_list_of_pairs

2.3 ANALYSIS

- *ADD*
  Add [with factors] two parameters of a detector an put result in a third one

  ADD P1 P2 P3 FACTOR f1 f2 OFFSET off3 GAIN g3

- *ADDBACK*
  Addback of composite detectors

  ADDBACK Dn

- *BANANA*
  One two-dimensional gate

  BANANA Px Py [In|Out] banana_file Rx Ry FOLD_GATE

- *BANANAS*
  Multiple two-dimensional gate

  BANANAS Px Py [In|Out] #bananas banana_file(#bananas_times) Rx Ry FOLD_GATE

- *COMBINE*
  Combine two parameters of a detector an put result in a third one

  COMBINE P1 P2 P3 LIMIT nchan

- *EBKILL*
  Kill detectors according to list of bad detectors

  EBKILL D bad_detectors.file [RUN|NORUN

- *FOLD*
  Discard event if number of detectors is outside limits

  FOLD D Min Max

- *GAIN*
  Change gain of a parameter

  GAIN Pn Offset Gain Wl Wh FOLD_GATE

- *GATE*
  One gate on a parameter

  GATE Pn [In|Out] Low High FOLD_GATE

- *GATES*
  Multiple gates on a parameter

  GATES Pn [In|Out] #gates (Low High)(#gates_times) FOLD_GATE

- *HK*
  Total energy H and fold K of a detector (e.g. BGO ball)

  HK Dn Fh Fk Offset Gain Wl Wh
  Offset, Gain, Wl, Wh applied to H

- *HSORT1D*
  One-dimensional sort of width 'cubic' or 'spherical' HGATES

  HSORT1D Px spectrumname [Res R] Hash [maxtimes>1] Cubic|Spherical

- *KILL*
  Kill detectors from the event

  KILL D {list_of_detectors_to_kill}

- *LIST_EVENT*
  List events on terminal or in a disk_file

  LIST_EVENTS [filename]

- *MASK*
  Binary mask of a parameter

  MASK Px mask

- *MERGE*
  Merge detectors together

  MERGE {list_of_detectors_type_to_merge_into_destination_D} D

- *MOVE*
  Move a list of detectors of one type into another type

  MOVE D [list_of_detectors] N [id_of_first] [GATE Pn [In|Out] Low High]

- *PIN*
  Particle Identification Number for the charged particles detected

  PIN Px Py Fn #bananas Rx Ry FOLD_GATE

  {#particles_in_banana_1 weight_of_banana_1 banana_file_1}

  {#particles_in_banana_2 weight_of_banana_2 banana_file_2}

  { .... }

- *PROJECTIONS*
  Projections for all defined parameters and detectors

  PROJECTIONS [Filename {for_all_defined_parameters}]

- *RECAL*
  Recalibration of a parameter. Coefficients from file

  RECAL Pn file.cal [RUN|NORUN] Offset Gain Wl Wh FOLD_GATE

- *RECAL_DOPPLER*
  Doppler correction with recoil velocity function of gamma energy

  RECAL_DOPPLER Pn v0_% [E0 E1 v1_%] (GASP|EB|Angles_file) Offset Gain Wl Wh FOLD_GATE

- *RECAL_LUT*
  Recalibration of a parameter from look_up table

  RECAL_LUT Pn file.lut [RUN|NORUN] Wl Wh FOLD_GATE

- *RECAL_KINE*
  Kinematic reconstruction of gamma ray energy parameter according to the geometry of the detected charged particles

  RECAL_KINE Pn (GASP|EB|Angles_file_Ge) Offset Gain Wl Wh FOLD_GATE

  Bx By #bananas Rx Ry description.file (GASP|EB|Angles_file_Si)

- *RECALL_EVENT*
  Recal the saved copy and continue analysis

  RECALL_EVENT ALWAYS or IFVALID

- *REORDER*
  Order the sequence of detectors of the event

  REORDER [D]

- *SELECT*
  Select events with defined detectors

  SELECT D {list_of_detectors_to_select}

- *SORT1D*
  One-dimensional sort of any parameter

  SORT1D Px spectrumname [Res R] [Hash [#times]]

- *SORT2D*
  Two-dimensional sort of any pair of parameters

  SORT2D Px Py matrixname [Res Rx Ry] [Step Sx Sy] [Hash [#times]]

- *SORT3D*
  Three-dimensional sort of any triplet of parameters

  SORT3D Px Py Pz matrixname [Res Rx Ry Rz] [Step Sx Sy Sz] [Hash [#times]]

- *SORT4D*
  Four-dimensional sort of any quadruplet of parameters

  SORT4D Px Py Pz Pt matrixname [Res Rx Ry Rz Rt] [Step Sx Sy Sz St] [Hash [#times]]

- *SORT2D_SYMM*
  Symmetrized two-dimensional sort

  SORT2D_SYMM Px matrixname [Res R] [Step S] [Hash [#times]]

- *SORT3D_SYMM*
  Symmetrized three-dimensional sort

  SORT3D_SYMM Px matrixname [Res R] [Step S] [Hash [#times]]

- *SORT4D_SYMM*
  Symmetrized four-dimensional sort

  SORT4D_SYMM Px matrixname [Res R] [Step S] [Hash [#times]]

- *SORT2D_HSYMM*
  Half-symmetrized two-dimensional sort

  SORT2D_HSYMM Px matrixname [Res R] [Step S] [Hash [#times]]

- *SORT3D_HSYMM*
  Half-symmetrized three-dimensional sort

  SORT3D_HSYMM Px matrixname [Res R] [Step S] [Hash [#times]]

- *SORT4D_HSYMM*
  Half-symmetrized four-dimensional sort

  SORT4D_HSYMM Px matrixname [Res R] [Step S] [Hash [#times]]

- *SORT3D_PAIR*
  A cube of Px-Py-Pair_index

  SORT3D_PAIR Px Py Pn matrixname [Res Rx Ry Rn] [Step Sx Sy Sn]

- *SPLIT* [1]
  Split detectors from one type to a list of ....

  SPLIT D {list_of_destination_types} {#detectors_for_each_destination}

- *STATISTICS*
  Calculate the statistics of detectors

  STATISTICS

- *STORE_EVENT*
  Save a copy of the event in its present status

  STORE_EVENT

- *SWAP*
  Swap two parameters

  SWAP Px Py

- *TIME_ADJUST*
  Improve the timming by adjustment of the time reference

  TIME_ADJUST Pn position rejection_factor Wl Wh FOLD_GATE

- *USERSUB*
  User defined routines

  USERSUB1
  ....
  USERSUB9

- <u>*WINDOW*</u>
  Gates on all the parameters of the defined type

  WINDOW P (Wl Wh){#parameters_times} FOLD_GATE

- <u>*WRITE_EVENT*</u>
  Write events to Tape or Disk_file (possibly in reduced format)

  WRITE_EVENT [Tape|Disk] [Reduce {0|1 for_every_defined_parameter}]

3. DETAILED DESCRIPTION OF THE SORTING COMMANDS

3.1 FORMAT

- *GASP*

  Defines the format of the event to be analized as being of the GASP type (cf. Appendix 1). This is the DEFAULT event format.

- *EUROBALL*

  Defines the format of the event to be analized as being of the EUROBALL type (cf. **EDOC312** PS file EUROBALL DAQ description).

- *HEADER*

  Defines the number of fixed parameters (**"#pars"**) present in the event (header of the event). The numbering of these parameters starts from 0 and further will be refered as **F0, F1, ...**. After **"#pars"** follows the list of the lengths of the spectra associated with the fixed parameters (**resolution**). For each defined fixed parameter it is associated a resolution **"Res"** number. In a standard (GASP I) measurement this command looks like:

  HEADER F 2 4096 4096

  and means that there are 2 fixed parameters each written on 4096 channels where F0 = sum energy from the BGO inner ball and F1 = fold spectrum from the BGO inner ball.

  There is the possibility to add fixed parameters to the header using the **"PLUS"** subcommand which allows to define **"#pars+"** supplimentary fixed parameters to be used during the analysis, each having specified the resolution.

  HEADER F 2 4096 4096 PLUS 2 512 4096

  In the above example two more parameters have been added (**F2 and F3**) with the resolutions 512 and 4096 channels, respectively.

  Maximum number of fixed parameters is 32.

- *DETECTOR*

  Defines the single-detector classes present in the event (**"D"**).
  For each class the command contains:

  number of detectors (**"#detectors"**)

  number of parameters for each detector (**"#pars"**)

  resolution of each of the parameters (**"#Res"**)

In a GASP event the Ge detectors are described as follows:

DETECTOR G 40 4 8192 4096 4096 4096

meaning that there are 40 Ge detectors each one having 4 parameters: G0 = energy written on 8192 channels, G1 = time written on 4096 channels, G2 = energy released in segment A and G3 = energy released in segment B.
For ISIS:

DETECTOR S 40 4 4096 4096 4096 4096

where: S0 = DE energy, S1 = E energy, S2 = DE time and S3 = E time.
For BGO inner ball:

DETECTOR B 80 2 4096 4096

where: B0 = energy, B1 = time.

One can add new parameters to the detector class using the **"PLUS"** subcommand which allows to define **"#pars+"** supplimentary parameters each having specified its resolution (see HEADER command). In this way one can define new detectors to be used during the analysis; e.g.:

DETECTOR T 20 PLUS 3 4096 4096 4096.

- *CDETECTOR*

Defines the composite-detector classes present in the event (**"C"**).
For each class the command contains:

number of detectors (**"#detectors"**)

number of segments (**"#segments"**)

number of parameters for each segment (**"#pars"**)

resolution of each of the parameters (**"#Res"**)

In an EUROBALL event the Ge Cluster detectors are described as follows:

CDETECTOR C 15 7 2 8192 8192

meaning that there are 15 cluster detectors each one having 7 segments with 2 parameters: C0 = energy, C1 = time written on 8192 channels.
For Ge Clover detectors:

CDETECTOR Q 26 4 2 8192 8192

For Ge Tapered detectors (if defined as composite as we locally prefer):

CDETECTOR T 30 1 2 8192 8192

One can add new parameters to the detector class using the **"PLUS"** subcommand which allows to define **"#pars+"** supplimentary parameters each having specified its resolution (see HEADER command).

3.2 DECLARATIONS

- *RAWFOLDMIN*

It acts when reading events from the tape selecting only the ones which have at least **"foldmin"** detectors. The minimum fold to be considered has to be defined for each defined detector in the order they have been defined in FORMAT section. Default value is 0.
It allows a fast selection of the events in terms of the multiplicity before analysing them.

RAWFOLDMIN 2 1
to accept only events which have at least 2 detectors from the first class and 1 from the second one.

- *HGATEDEF*

It defines a number **"#gates"** of gates to be applied on the parameter **"Pn"**. Wl and Wh define the limits of each gate. Gates are specified in separate lines.
The gates can be specified also from a file **"file_with_gates"**. The gates can be specified for each detector individually (**"Individual"**) or the same for all of them (**"Same_for_all"**). The limits of the gates have to be specified in channels. The channels Wl and Wh belong to the gate.

It is used in order to create cubes, matrices or spectra in coincidence with gates on a specified parameter.

The file with gates has the following format:

```
Case 1: "Individual"
   ADC 00      |
   Wl1   Wh1   |
   Wl2   Wh2   |
   ...........  |
   ADC 01      |
   Wl1'  Wh1'  |
   Wl2'  Wh2'  | ====> individid.gates
   ...........  |
   ...........  |
   ...........  |
   ADC 39      |
   Wl1"  Wh1"  |
```

Wl2" Wh2" |

.... |

Case 2: "Same_for_all"

Wl1 Wh1

Wl2 Wh2

....

HGATEDEF G1 individ.gates INDIVIDUAL

meaning that the gates on the parameter **G1** are listed in the file **individ.gates** for each detector separately.

- *PAIRDEF*

3.3 ANALYSIS

- *ADD*

Gives the possibility to sum two parameters, **"P1"** and **"P2"**, and to put the result in a third one, **"P3"**, all of them belonging to the same type of detectors. All three parameters have to be previously defined. The parameters to be added must have the same dimension. The operation can be done also using multiplicative factors for each of the three parameters, **"f1"**, **"f2"** and **"f3"**, respectively. The DEFAULT value for the factors is 1. They cannot be 0.

DETECTOR G 40 4 8192 4096 4096 4096 PLUS 1 4096

ADD G2 G3 G4 1 1 2

adds parameters G2 and G3 the result being stored in G4 after multiplying the result by 2.

- *ADDBACK*

Performs the addback of the energy signals in the segments of composite detectors. The procedure is applied when 2 or 3 neighbouring segments of the same composite detector fired in coincidence.

Cluster detectors case:
- all the possible combinations of two neighbouring segments; when the two segments are not neighbouring the signals are treated as two different events;
- combinations of two neighbouring segments plus the central one.

Clover detectors case:
-combinations of two neighbouring segments excluding the diagonal cases.

In all cases of addback the sum energy is attributed to the segment in which the major energy was released.

ADDBACK G1

The addback procedure will be applied for the G1 parameter.

- *BANANA* [2,3]

Defines a two-dimensional gate in the plane specified by the parameters **"Px"** and **"Py"**. The command is selecting only the events having the pair of parameters inside the banana gate. The dimension of space in which the banana was defined is given by the resolution parameters **"Rx"** and **"Ry"**. The points defining the banana gate are taken from the file **"banana.file"** where they are listed on two columns X (Px) and Y (Py).

BANANA S1 S0 banana.file 1024 1024 1 20

selects the events in which the S1 S0 coincidences are inside the surface defined by the banana file "banana.file", considering 1024 channels for each parameter, and only if the condition is satisfied by at least one pair and less than 21.

- *BANANAS* [2,3]
  Defines more bananas in the space defined by the same parameters in an OR relationship.

  BANANAS S1 S0 OUT 2 ban1.file ban2.file 1024 1024 1 20
  only events which have the pair (S1,S0) outside the two bananas are considered.

- *COMBINE*

  Merges the first two parameters, **"P1"** and **"P2"**, in a third one, **"P3"** (all of them belonging to the same type of detectors). All three parameters have to be previously defined. The resulting spectrum will be built as follows:
  P1 if its content is lower than **"nchan"**
  if its content is higher than **"nchan"** but P2 is missing
  P2 if its content is higher than **"nchan"**

  DETECTOR G 239 3 8192 8192 8192 PLUS 1 8192

  COMBINE G1 G0 G3 LIMIT 7000
  combine the 4 MeV and 20 MeV parameters (G1 and G0) the result being stored in G3. The 4 MeV data are taken below channel 7000 (e.g., 3.5 MeV if data are calibrated to 0.5 keV/ch) and the 20 MeV data above channel 7000.

- *COPY*

- *EBKILL*
  Allows to eliminate the bad detectors for all runs or for some of them. In the case of composite detectors the detector is throuwn away only if the bad capsule or one of its neighbours was hit. The list of bad detectors is organized as follows:

  RUN 1

  12 14 90 189

  RUN 6

  12 84 90 208

  RUN 10

  12

  ....

  For the NORUN case the file contains only one row with the numbers of the bad capsules.

  EBKILL G detector.bad RUN
  detectors G will be eliminated from the events according to the list "detector.bad" which is organized in a RUN dependend manner.

- *FOLD*

  It is used in order to put a window on the number of detectors of class **"D"** to be considered for the analysis. The window is defined by the numbers **"Min"** and **"Max"**. By DEFAULT the fold window is completely opened.

FOLD G 2 40

means that only events having at least 2 and less than 41 **G** type detectors fired are accepted.

- <u>*GAIN*</u> [2]

  Changes the gain of one parameter (**"Pn"**). The linear recalibration is done according to the coefficients **"Offset"** and **"Gain"**. A gate is defined on the recalibrated parameter (**"Wl"** and **"Wh"**). The event is passed to the program for further processing only if the **"FOLD_GATE"** condition is satisfied.

  GAIN G0 0 2 10 2047 2 20

  changes the gain of the G0 parameter from (OLDGAIN) to (OLDGAIN/2) and keeps only events having the recalibrated parameter between 10 and 2047 and there at least two of themin the event and less than 21.

- <u>*GATE*</u> [2,3]

  Defines a gate on one of the parameters (**"Pn"**). If the parameter does not satisfy the condition the event is discarded.

  GATE F0 10 4095

  put a gate on the F0 header parameter between the channels 10 and 4095.

  GATE G0 10 4095 2 20

  put a gate on the G0 detector parameter between the channels 10 and 4095 if the **"FOLD_GATE"** condition is satisfied.

- <u>*GATES*</u> [2,3]

  Defines a number of **"#gates"** gates to be applied on the **"Pn"** parameter. The gates are in and OR relationship. If the parameter does not satisfy the condition the event is not analysed.

  GATES F2 3 100 120 200 220 300 320

  means that the event is valid only if the HEADER paramter F2 is inside the limits of one of the three defined gates.

  GATE G1 OUT 2 10 100 1000 4095 2 20

  a valid event has at least 2 but less than 20 G1 parameters outside the limits of the two defined gates.

  This command is implemented through a look-up table and therefore a check is performed that gate limits are consistent with the resolution of the parameter as defined in the SETUP file.

- <u>*HK*</u>

  Builds the sum energy and multiplicity spectra for the parameter **"Dn"** of a class of detectors. The spectra are written in the HEADER type parameters **"Fh"** and **"Fk"**, respectively. These parameters have to be defined previously in the HEADER command.

The sum energy spectrum is compressed by a a factor of **"Gain"** and shifted with the **"Offset"** value.The final spectrum is cut between the channels **"Wl"** and **"Wh"**.

As example can be the case when the BGO detectors of the GASP inner ball are recorded as individual detectors:

HEADER F PLUS 2 2048 128

DETECTOR B 80 2 4096 4096

HK B0 F0 F1 0 2 10 2047

The sum energy sum and multiplicity of the inner ball are reconstructed from the individual detector information and put in the HEADER parameters F0 and F1, respectively. The sum energy is compressed by a factor 2 and cut between the channels 10 and 2047. The multiplicity spectrum is recorded on 128 channels.

- *HSORT1D* [6]

Produces 1D spectra of the parameter **"Px"** named **"spectrumname"** with the resolution **"R"**. DEFAULT value for the resolution is the resolution of the parameter. The spectrum file contains a stack of spectra from #00 to #**"maxtimes"** meaning that the first one (#00) is in coincidence with zero gates, the second one (#01) is in coincidence with one of the gates from the list, the third one (#02) is in coincidence with two of the gates from the list and so on.

The gates can be of **S** type (**"Spherical"**) or of **C** type (**"Cubic"**).

HSORT1D G0 MULTIPLE.GATED 4096 Hash 9 S In the file MULTIPLE.GATED 10 spectra of the G0 parameter 0.1,2,...,9 times gated are written each one having a length of 4096 channels. The gates are treated as spherical ones.

- *KILL*

Eliminates from the event the detectors of **"D"** type specified in **"list_of_detectors_to_kill"** by their ADC numbers.

KILL G 0 1 2 3 4 5

eliminates from the event the G type detectors from 0 to 5.

- *LIST_EVENT*

Lists the content of the events on terminal or disk_file (**"filename"**) in a decoded form.

F 403 409

G 2 32 1834 2452 0 0 17 1470 890 1120 350

S 2 6 135 551 603 448 8 70 444 818 734

meaning that:

$$
\begin{aligned}
\text{bgo\_esum} &= 403 \\
\text{bgo\_mult} &= 409 \\
\text{no\_of\_fired\_Ge} &= 2 \\
\text{id} &= 32 \\
\text{ener} &= 1834 \\
\text{time} &= 2452 \\
\text{ener\_A} &= 0 \\
\text{ener\_B} &= 0 \\
\text{id} &= 17 \\
\text{ener} &= 1470 \\
\text{time} &= 890 \\
\text{ener\_A} &= 1120 \\
\text{ener\_B} &= 350 \\
\text{no\_of\_fired\_Si} &= 2 \\
\text{id} &= 6 \\
\text{de} &= 135 \\
\text{e} &= 551 \\
\text{tde} &= 603 \\
\text{te} &= 448 \\
\text{id} &= 8 \\
\text{de} &= 70 \\
\text{e} &= 444 \\
\text{tde} &= 818 \\
\text{te} &= 734
\end{aligned}
$$

- *MASK*

- *MERGE*

  Starting from several classes of detectors forms a unique list of detectors ordered by id's in a new class of detector defined to have the proper number of detectors.

  CDETECTOR C 15 7 3 8192 8192 8192

  CDETECTOR Q 26 4 3 8192 8192 8192

  CDETECTOR T 35 1 3 8192 8192 8192

  DETECTOR G 244 PLUS 3 8192 8192 8192

  MERGE C Q T G

  merges together the composite detectors C, Q, T in the G detector class. The mapping of the input list of detectors (C, Q, T) in the output one (G) is the following:

G#000	C#00	Seg#0
G#001		Seg#1
.....		.....
G#006		Seg#6
G#007	C#01	Seg#0
G#008		Seg#1
.....		.....
G#013		Seg#6
.....	....	.....
G#105	Q#00	Seg#0
.....		.....
G#108		Seg#3
G#109	Q#01	Seg#0
.....		.....
G#112		Seg#3
.....	....	.....
G#209	T#00	Seg#0
G#210	T#01	Seg#0
.....	....	.....
G#243	T#34	Seg#0

- *MOVE* [3]

Moving a list of detectors of one type into another type. The new type of detector has to be previously defined. The action take place only if the gate on the parameter **"Pn"** is satisfied.

HEADER F 2 4096 4096

DETECTOR G 40 4 8192 4096 4096 4096

DETECTOR S 8 PLUS 4 8192 4096 4096 4096

MOVE G 16 17 18 19 20 21 22 23 S 0 GATE F1 300 4095

detectors G with id's from #16 to #23 are moved in the S new type of detector with the id's from #00 to #07 only if the F1 parameter has a value between 300 and 4095.

- *PIN*

Creates a new HEADER parameter **"Fn"** containing a multiplicity spectrum built by counting all the pairs of (**"Px", "Py"**) parameters, with the right fold (**"FOLD_GATE**), inside the bananas. The fixed parameter Fn has to be defined previously as a HEADER parameter with the **PLUS** subcommand. Up to four ("#bananas") bananas can be defined in the same (x,y) space of **"Rx"** and **"Ry"** dimensions, respectively. The **"#particles_in_banana_n"** means that the pair (**"Px", "Py"**) in the **"banana_file_n"** counts for **"#particles_in_banana_n"** particles (number of simultaneous hits in the same detector). The **"weight_of_banana_n"** specifies the rule according to which the spectrum is organized.

HEADER F 2 4096 4096 PLUS 1 64

PIN S1 S0 F2 3 1024 1024 1 20

x a 1P.BAN

y a 2P.BAN

z b 1A.BAN

the multiplicity spectrum is incremented in the F2 parameter defined on 64 channels. The 1P.BAN and 2P.BAN bananas count for x and y simultaneously detected P type particle (of the same kind) in the same detector, respectively. The 1A.BAN counts for z simultaneously detected A type of particles in the same detector.

The spectrum is built following the rule:

$$\{\mathbf{x} * (\mathbf{\#pairs\_in\_banana\_1P}) + \mathbf{y} * (\mathbf{\#pairs\_in\_banana\_2P})\} * \mathbf{a}$$
$$+$$
$$\{\mathbf{z} * (\mathbf{\#pairs\_in\_banana\_1A})\} * \mathbf{b}$$

- *PROJECTIONS*

Produces the projection spectra corresponding to the definition of the event structure including eventually recalibration or gates of the parameters. The names for the projection files can be explicitly given; if not the DEFAULT names derived from the name of detectors are used. The spectra are saved at the end of each RUN and the number of RUN is specified in the extension of the file name (in I4 format). They are written in L format (4 bytes integers). The files are organized as libraries of spectra with the length given by the resolution of each parameter. The number of spectra is defined by the number of the detectors. The spectra associated with the HEADER parameters are packed inside a unique file with the resolution defined for each of them. Various programs for data analysis (TRACK, SADD) are able to extract the spectra from these libraries. The spectrum associated with G0 parameter for the 21st detector measured during RUN#20 can be specified as: EGE#20.0020/L:8 (in format L on 8K).

PROJE FIX EGE TGE AGE BGE EDE EEE TDE TEE

to create the projections for a standard GASP event.

- *RECAL* [2]

Defines the recalibration of one of the parameters. The calibration coefficients (not necessarily linear calibration) are taken from the file **"file.cal"**. The coefficients are read one by one at the beginning of the analysis. They can be RUN dependent (DEFAULT) or RUN independent (option NORUN). The structure of such a file is the following:

RUN# ADC# n+1 C0 C1 C2 ... Cn

where C0,..., Cn are the calibration coefficients and n is the order of the calibration polynomial (maximum 4). Once the recalibration done a linear alignment of the gains (defined by **"Offset"** and **"Gain"**) is performed. These coefficients must be present in the command line even if the readjustment is not desired (case in which they have the values 0 and 1, respectively). Then the gate **"Wl" " Wh"** is applied to the recalibrated data. Also a multiplicity selecton is done by the **"FOLD_GATE"**.

In the case of a HEADER parameter if the gate condition is not satisfied the event is thrown away. In the case of DETECTOR type parameters only the detector is eliminated.

RECAL G0 Ge_ener.cal 0 2 10 2047 2 20
this command defines the recalibration ofthe G0 parameter (e.g., energy of the GASP Ge detectors) using the coefficients specified in the file Ge_ener.cal for each RUN. Once recalibrated the parameter is amplified by a factor 2 resulting in a dispersion of 0.5 keV/channel; the detector is considered only if the result fits in the energy range from 10 to 2047 (including the two limits); the event is considered only if at least two detectors satisfy this condition and less than 21.

- *RECAL_DOPPLER* [2]

Produces a Doppler correction of the gamma energy with the velocity value depending on the gamma energy. If **"E0 E1 v1_%"** are specified then from 0 up to **"E0"** the **"v0_%"** velocity is used, from **"E1"** to the maximum resolution the **"v1_%"** velocity is used and between the **"E0"** and **"E1"** values a linear interpolation is done for the velocity as a function of the channel.
By vn_% one means v/c in percentage.
The detection angles can be read from standard GASP/EB files or from a user defined file **"Angles_file"** which has to be written in the format:

ADC#00	θ angle	ϕ angle
ADC#01	θ angle	ϕ angle
....	....	....

RECAL_DOPPLER G0 2.5 1000 1500 3.0 GASP 0 1 10 2047 2 20
meaning that the G0 parameter will be Doppler corrected with v/c=2.5% form 0 t0 1000, with v/c=3.0% from 1500 to maximum value in-between being used a linear interpolation between the two v/c limits. Angles are taken from the standard GASP angles configuration file. Data are calibrated to 1 keV/ch and the event is valid only if at least 2 and less than 20 detectors have the new G0 value between 10 and 2047 keV

- *RECAL_LUT* [2]

Performs the recalibration of a parameter according to a look-up table (**"file.lut"**).

RECAL_LUT F1 BGO.LUT LUT 0 1 10 4095
create a spectrum of the F1 parameter according to a rule define in the BGO.LUT file. A LUT file of the form:

10

20

30

.....

will produce a spectrum in which the channels from 0 to 9 will be mapped into the 0 channel; the channels from 10 to 19 will be mapped into the channel 1, and so on.

This command is used generally for the F1 header parameter (the inner ball fold distribution) and in order to map the wires on the X axis of the RMS (defined as a fixed parameter in the header).

- <u>*RECAL_KINE*</u> [2]

Performs a Doppler correction of the gamma ray energy parameter **"Pn"** with the recoil velocity and direction determined according to the energies and the angles of the detected charged particles. The position of the Ge detectors and of the charged particles detectors are taken from standard files **"GASP/EUROBALL"** or from userdefined files **"Angles_file_Ge / Angles_file_Si"**. Data are recalibrated according to the parameters **"Offset"** and **"Gain"** and cut between the limits **"Wl"** and **"Wh"**. Events are considered only if **"FOLD_GATE"** is satisfied. The informations regarding the energy distribution of the charged particles on the various detectors and the reaction kinematics are given in the file **"description.file"**. This file file is organized as follows:

First row:	M^{NC} and E^{CM}
Second row:	$m^{particle}$
Third row:	N (number of detectors for which the energy is given: from ADC#00 to ADC#(N-1))
Fourth row:	the values of the particles energy
Fifth row:	(a number of N values). Detectors with
....	zero energy released are not specified
n-th row:	name of the file containing the points of the
particle banana	

The rows sequence from 2 to n is repeated for each emitted particle in order to populate the reaction channel of interest (**"#bananas"** times).

The banana gates have been defined in the space (**"Bx"**,**"By"**) with the resolution (**"Rx"**,**"Ry"**)

For the case of the ISIS detector mounted inside GASP for an $\alpha 2p$ reaction channel we will have:

RECAL_KINE G0 GASP 0 2 10 2047 2 20

S1 S0 3 1024 1024 a2p.set GASP

The command acts on the G0 Ge energy parameter. The energy spectrum will be compressed two times and cut between the channels 10 to 2047. Valid events are those which have at least 2 and less than 21 fired Ge detectors with the right G0 parameter.

The a2p.set description file has the following format:

56 71.4

4

16

22.1 22.1 23.6 23.6 22.1 22.1 13.1 13.1 13.1 13.1

13.7 13.7 9.1 9.1 9.1 9.1

alpha.ban

1

32

9.4 9.4 9.8 9.8 9.4 9.4 6.6 6.6 6.6 6.6 6.8 6.8 5.3

5.3 5.3 5.3 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 2.5 2.5

2.5 2.5 2.0 2.0 2.0 2.0

proton.ban

1

32

9.4 9.4 9.8 9.8 9.4 9.4 6.6 6.6 6.6 6.6 6.8 6.8 5.3

5.3 5.3 5.3 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 2.5 2.5

2.5 2.5 2.0 2.0 2.0 2.0

proton.ban

The bananas alpha.ban and proton.ban have been defined in the space E vs. ΔE (S1,S0) reduced to 1024 x 1024 channels.

- *RECALL_EVENT*

  Recalls the event in the status it was saved with STORE_EVENT command.

- *REORDER*

  Produces a reordering of the detectors in the event.

- *TIME_ADJUST*
  Improve the timming by adjustment of the time reference.

  TIME_ADJUST G2 1000 0.95 10 6000 1 100

- *SELECT*

  Only events which contain the **"D"** type detectors listed in **"list_of_detectors_to_select"** are analized (all the others being rejected).

  SELECT G 0 1 2 3 4 5
  to select only the events with the G type detectors from 0 to 5.

- *SORT1D* [5]

  Produces a one-dimensional spectrum of the **"Px"** parameter. The spectrum will have the name **"spectrumname"**. DEFAULT resolution for such a spectrum is given by the maximum resolution of the parameter. To use a different resolution one has to specify it through the **"Res"** argument **"R"** and **"Ry"**.

SORT1D G0 G0.SPE Res 2048
to create a 1D spectrum with the name G0.SPE containing the G0 parameter.

- *SORT2D, SORT3D, SORT4D* [4,5]

Produces a 2/3/4-dimensional matrix between the **"Px/Py/Pz/Pt"** parameters. The matrix will have the name **"matrixname"** with the standard extension **CMAT** and will be written in compressed format. DEFAULT resolution for such a matrix is given by the maximum resolutions of the parameters. To use a different resolution one has to specify it through the **"Res"** arguments **"Rx/Ry/Rz/Rt"**. All axes has to be specified. The standard dimension of the 2/3/4-D segments is (64 x 64 / 64 x 64 x 64 / 64 x 64 x 64 x 64). They can be changed through the **"Step"** arguments **"Sx/Sy/Sz/St"**. The length of each axis has to be an integer number of the steps.

SORT2D S1 S0 EDE Res 1024 1024 Step 32 32
to create a 2D matrix with the name EDE.CMAT having on the first axis the S1 parameter and on the second one the S1 parameter.

SORT3D F2 F4 G0 RMS Res 512 512 2048 Step 64 64 32
to create a 3D matrix with the name RMS.CMAT having on the first axis the F2 parameter, on the second one the F3 parameter and on the third one the G0 parameter.

SORT4D F0 F1 F2 G0 HKMG Res 512 126 512 1024
to create a 4D matrix with the name HKMG.CMAT having on the first axis the F0 parameter, on the second one the F1 parameter, on the third one the F2 parameter and on the fourth axis the G0 parameter.

- *SORT2D_SYMM, SORT3D_SYMM, SORT4D_SYMM* [4,5]

Produces a 2/3/4-dimensional symmetrized matrix for the **"Px"** parameter. DEFAULT values for the resolution (**"R"**) and step (**S"**) are 4096 and 64. The matrix will have the name **"matrixname"** with the standard extension **CMAT** and will be written in compressed format.

For 2D matrices:
the matrix is symmetrized with the condition ind2≥ind1 when the (ind1,ind2) location is incremented (with an eventual permutation of the two index). It results in a reduction by a factor two of the dimension of the matrix (the dimension of the matrix is reducing from res*res to C(res+1,2)=res*(res+1)/2. Due to the segmentation of the matrix the number of segments is changing from $(\text{res/step})^2$ (4096 in the standard case) to C(res/step+1,2) (tipically 2080).

HGATEDEF G0 5

150 160

250 260

350 360

450 460

550 560

GSORT2D_SYMM G0 MAT2D Hash 2

The matrix is incremented by double gating on the G0 parameter with the list of gates given in the HGATEDEF command.

For 3D matrices:
the matrix is symmetrized with the condition ind3≥ind2≥ind1 when the (ind1,ind2,ind3) location is incremented. It results in a reduction by a factor six of the dimension of the matrix. The standard dimension for each axis is 2048 and is divided in 2048/32=64 portions. The total number of segments in which the cube will be divided is C(64+2,3)=45760.

GSORT3D_SYMM G0 MAT3D

For 4D matrices:

GSORT4D_SYMM G0 MAT4D

- *SORT2D_HSYMM, SORT3D_HSYMM, SORT4D_HSYMM* [4,5]

Produces half-symmetrized 2/3/4D matrices to allow fast access to the data stored inside.

SORT3D_HSYMM G0 HMAT3D

- *SORT3D_PAIR*

Creates a Px-Py-Pindex cube. The index value for each valid pair of detectors is specified in the "file_with_index_ of_pairs" written in the form:

ADC#1 ADC#2 index

....

In the GASP directory there are available two files for such a sort type:

GASP-PAIRS.LST	7 matrices of the type all_the_detectors against the detectors_at_one_angle
GASP-PLUNGER.LST	28 matrices of the type detectors_at one_angle against detectors_at_another_angle.

used for angular distribution and angular correlation analysis.

PAIRDEF GASP-PAIRS.LST

PAIRDEF GASP-PLUNGER.LST

....

SORT3D_PAIR G0 G0 P0 PAIRS RES 4096 4096 32 STEP 64 64 8

SORT3D_PAIR G0 G0 P1 PLUN RES 4096 4096 32 STEP 64 64 8

to create:

1. a stack of 7 matrices of angular distribution indexed by P0.
2. a stack of 28 matrices of plunger type indexed by P1.

- *SPLIT*

- *STORE_EVENT*

  Saves internally a copy of the event in its status at the moment when the command is given.

- *STATISTICS*

  Calculates the statistics of detectors.

- *SWAP*

  Interchanges the values of the parameters **"Px"** and **"Py"**

- *TIME_ADJUST*

  Improve the timming by adjustment of the time reference. The operation consists in calculating the centroid of the time distribution in one event and to recalculate the time positions with respect to this position. Times which are outside a number of **"rejection_factor"** of sigma are not considered in building the final centroid position. Before using this command one has to recalibrate the centroids of the time distributions of all the detectors at the same position (using eventually the RECAL_TIME program).

  TIME_ADJUST G2 2000 2 1900 2100 2 50

  time spectra recorded on G2 parameter are adjusted to bring the centroid of the time distribution at position 2000. Times which are more than 2 sigma away from the centroid are not considered in calculating the time distribution. Finally, only detectors which have G2 from 1900 to 2100 and multiplicity from 2 to 50 are considered.

- *WINDOW* [2]

  Defines gates on all the parameters of the specified type (**"P"**). All the gates has to be satisfied in order to declare the event valid.

  WINDOW F 320 4095 220 4095

  gates have been set on all the HEADER type parameters F. Gates are set on the first two parameters (F0 and F1). The rest of them (#pars-2) are completely opened (from 0 to maximum dimension).

- *WRITE_EVENT*

  Causes the program to write the events on tape (**T**) or the disk (**D**) in their form at the moment when the command is specified according to the GASP standard format with the possibility of eliminating the parameters not needed (**"REDUCE"**). By DEFAULT events are written on tape. When REDUCE option is used for each defined parameter has to be specified **"1"** if it has to be preserved or **"0"** if it has to be eliminated (all the parameters have to be in the list).

  HEADER F 2 4096 4096

  DETECTOR G 40 4 8192 4096 4096 4096

WRITE_EVENTS D REDUCE 1 1 1 0 0 0

writes the list of events on a disk file in reduce format keeping the F0, F1 and G0 parameters.

[1]Not yet defined.

[2]For HEADER type parameters the **"FOLD_GATE"** subcommand is not effective and needs not to be specified.

[3]The **"IN|OUT"** subcommand means that the parameter has to be inside/outside the gate limits. DEFAULT is IN.

[4]The 4D sort is not yet available.

[5]If present the **"Hash"** subcommand means that the spectrum is incremented in coincidende with the gates defined by HGATEDEF command applied **"#times"** times.

[6]Subjected to changements.

Appendix 1

GASP Standart Event Format

The data files are written with fixed record length which usually is 32k (1k = 1024bytes). Each record has an 16 words (1word = 2bytes) header for informations. The structure of the header is the following:

typedef struct {

short int	rec_k_length	record length in k
unsigned short int	rec_number	record number; 0 for header and trailer
unsigned short int	run_number	run number
char	rec_id[2]	HG header; DG data; TG trailer
short int	header_len	header length in words (16)
short int	tape_num	tape number
short int	tape_part	part 0|1 if buffer ping-pong else 0
short int	data_source	data path 1=raw, 2=recal, 3=filter
long int	byte_order	=0xff00f00f for swapping problems
char	gasp_string[6]	="GASP" to identify histograms

} tape_record_head;

The byte_order variable can be used for automatic byte and/or word swapping. The byte order on the GASP tape format is the DIGITAL one.

Each file has a header record (rec_id="HG") with additional informations staring from word 16:

typedef struct {

char	beam_time_lid[17]
char	measure_name[17]
char	measure_comment[73]
char	run_number[7]
char	run_comment[73]
char	tape_number[7]
char	tape_part[3]
char	date_time[25]

} tape_header_com;

Beginning from word 160 a copy of the NEO acquisition program is listed in UNIX like format (single string with lines ended by LF). If the length of the program exceeds the record length extra records are written to accomodate the whole program.

In the data records (rec_id="DG") events are packed together padded with zeroes at the end of the record (no broken events). Events begin with a negative word 0xfnnn where nnn equal to the length in words of the event (excluding count word).

	label	EVENT_CLASSIFICATION	
	tag	RECORD_TAG	
	bgo_esum	4096	
	bgo_mult	4096	
	.		
	.	here other FIX params	
HEADER	.	can be added	
	.		
	ge_count	COUNTER ge 40	(number of Ge type detectors)
	ge_start	POINTER ge	
	si_count	COUNTER si 40	(number of Ge type detectors)
	si_start	POINTER si	

	id	KEY (the ADC number)	|
	ener	8192	| repeated for each
DETECTOR ge	time	4096	| fired Ge detector
	ener_A	4096	|
	ener_B	4096	|
	......	room for more params	|

	id	KEY (the ADC number)	|
	de	4096	| repeated for each
DETECTOR si	e	4096	| fired Si detector
	tde	4096	|
	te	4096	|
	...	room for more params	|

The last record of each file has a rec_id="TG" ad contains the same information as the header record except for the listing of the NEO program. It is used mainly as a time stamp of the end of the run.

Example of a hexadecimal dump of one event:

	f01b		Start event & length (27 words)
0	0000	label	
1	0000	tag	
2	0101	bgo_esum	
3	0328	bgo_mult	
4	0003	ge_count	3 Ge fired
5	0008	ge_start	start at word 8
6	0002	si_count	2 Si telescopes fired
7	0011	si_start	start at word 8(ge)+11-1 = 17
8	0017	ge.id	first Ge ADC#23
9	051e	ge.ener	
10	0700	ge.time	
11	0003	ge.id	second Ge ADC#03
12	12f0	ge.ener	
13	05ee	ge.time	
14	0002	ge.id	third Ge ADC#02
15	032a	ge.ener	
16	05ef	ge.time	
17	0001	si.id	first Si ADC#01
18	00aa	si.de	
19	0152	si.e	
20	076c	si.tde	
21	0738	si.te	
22	0005	si.id	second Si ADC#05
23	0202	si.de	
24	0281	si.e	
25	06c5	si.tde	
26	0675	si.te	
	fnnn		Start next event

Appendix 2

Examples of SETUP files

- Reducing EB data format to GASP data format

```
EUROBALL
CDETECTOR C 15 7 3 8192 8192 8192
CDETECTOR Q 26 4 3 8192 8192 8192
CDETECTOR T 30 1 3 8192 8192 8192
DETECTORS S 40 4 4096 4096 4096 4096
DETECTORS G 239 PLUS 3 8192 8192 8192
MERGE C Q T G
```

- Sorting DCO matrices

```
DETECTORS G 40 4 8192 4096 4096 4096
DETECTORS A 8 PLUS 4 8192 4096 4096 4096
DETECTORS B 12 PLUS 4 8192 4096 4096 4096
RECAL G1 GE_TIME.CAL 0 1 1950 2050 2 20
RECAL G0 GE_ENER.CAL 0 2 10 4095 2 20
MOVE G 16 17 18 19 20 21 22 23 A 0
MOVE G 0 1 2 3 4 5 34 35 36 37 38 39 B 0
SORT2D A0 B0 DCO.CMAT RES 4096 4096
```

- Sorting angular distributions cubes

```
HEADER F 5 4096 4096 4096 4096 4096
DETECTOR G 40 4 8192 4096 4096 4096
PAIRDEF PAIRS.LST
RECAL G1 GE_TIME.CAL RUN 0 1 950 1050 2 20
RECAL G0 GE_ENER.CAL RUN 0 2 10 4096 2 20
SORT3D_PAIR G0 G0 P0 DCO3D RES 4096 4096 128 STEP 128 128 8
```